



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Doble grado en ingeniería
eléctrica y mecánica

Trabajo Final de Grado

Estudio de las etapas de automatización de un proceso industrial y sus implicaciones en la gestión de la producción

Victor Balasch Rodríguez

Director: Miguel Delgado Prieto

- Departamento de electrónica

Directora: Elena Martí Badia

- Departamento de organización de empresas

10 DE JUNIO DE 2019

ÍNDICE

1. INTRODUCCIÓN	7
1.1 OBJETIVO.....	7
1.2 REQUISITOS.....	7
1.3 ANTECEDENTES.....	7
1.4 ALCANCE DEL ESTUDIO.....	9
2. DESCRIPCIÓN DEL PROCESO.....	10
3. MARCO TEÓRICO SOFTWARE DE SIMULACIÓN DE PROCESOS INDUSTRIALES	12
4. MARCO TEÓRICO AUTOMATIZACIÓN DE PROCESOS INDUSTRIALES	16
4.1 PLC O AUTÓMATA PROGRAMABLE	17
4.2 CICLO DE SCAN.....	21
4.3 DETECTORES Y ACTUADORES	22
4.4 PROGRAMACIÓN DEL PLC	23
4.4.1 Contexto histórico	23
4.4.2 Lenguajes de programación.....	24
4.4.2.1 Lenguaje Ladder	25
4.4.2.2 Texto estructurado	28
5. DISEÑO Y OPTIMIZACIÓN DEL PROCESO INDUSTRIAL.....	30
5.1 METODOLOGÍA.....	30
5.2 CONSTRUCCIÓN Y PARAMETRIZACIÓN DEL MODELO DE SIMULACIÓN.....	32
5.3 DEFINICIÓN DE MÉTRICAS	46
5.4 ESTUDIO DE LA FLEXIBILIDAD DEL SISTEMA ESCOGIDO.....	48
6. IMPLEMENTACIÓN DE LA SOLUCIÓN ÓPTIMA	49
6.1 DESCRIPCIÓN DEL SISTEMA EXPERIMENTAL	49
6.2 DIAGRAMA DE FLUJO DEL PROCESO	52
6.3 CONFIGURACIÓN DEL PROGRAMA	55
6.4 ESTRUCTURA DEL PROGRAMA.....	57
7. CONCLUSIONES	64
8. PRESUPUESTO	65
9. BIBLIOGRAFÍA.....	66

Tabla de Ilustraciones

ILUSTRACIÓN 1: ESQUEMA DE PROCESO.....	11
ILUSTRACIÓN 2: COMPARATIVA DE SOFTWARES DE SIMULACIÓN DE PROCESOS INDUSTRIALES	15
ILUSTRACIÓN 3: PLC NANO	17
ILUSTRACIÓN 4: PLC COMPACTO	18
ILUSTRACIÓN 5: PLC MODULAR.....	18
ILUSTRACIÓN 6: CICLO DE SCAN.....	21
ILUSTRACIÓN 7: DETECTOR INDUCTIVO CON RETENEDOR.....	22
ILUSTRACIÓN 8: OSCILACIÓN ONDA DETECTOR INDUCTIVO.....	22
ILUSTRACIÓN 9: ELECTROVÁLVULA NEUMÁTICA	23
ILUSTRACIÓN 10: TABLA LENGUAJE LADDER	27
ILUSTRACIÓN 11: EJEMPLO LADDER.....	27
ILUSTRACIÓN 12: EJEMPLO TEXTO ESTRUCTURADO	29
ILUSTRACIÓN 13: ESQUEMA PROCESO DE MODELIZACIÓN PARA LA SIMULACIÓN DE UN PROCESO INDUSTRIAL	31
ILUSTRACIÓN 14: ESQUEMA DE PROCESO CON PUNTO GESTIÓN PRIORIDAD.....	32
ILUSTRACIÓN 15: ESQUEMA DE PROCESO CON COLAS PA Y PT	33
ILUSTRACIÓN 16: PROCESO COMPLETO DE LA SIMULACIÓN EN ARENA	34
ILUSTRACIÓN 17: MÓDULO DE CREACIÓN DE ENTIDADES.....	35
ILUSTRACIÓN 18: MÓDULO ASIGNACIÓN DE ATRIBUTOS 1	35
ILUSTRACIÓN 19: MÓDULO PRIORIDAD.....	36
ILUSTRACIÓN 20: MÓDULO ASIGNACIÓN DE ATRIBUTOS 2	36
ILUSTRACIÓN 21: CONFIGURACIÓN COLA TALADRADO 1	37
ILUSTRACIÓN 22: CONFIGURACIÓN COLA TALADRADO 2	37
ILUSTRACIÓN 23: MÓDULO DEMORA	37
ILUSTRACIÓN 24: TABLA DEMORAS.....	38
ILUSTRACIÓN 25: MÓDULO DE SEPARACIÓN DE PIEZAS NANR DEL RESTO	38
ILUSTRACIÓN 26: MÓDULO DE SEPARACIÓN DE PIEZAS SANR DE CORRECTAS	39
ILUSTRACIÓN 27: MÓDULO AGUJEREADO.....	40
ILUSTRACIÓN 28: MÓDULO TALADRADO	41
ILUSTRACIÓN 29: MÓDULO COLA POSTAGUJEREADO2.....	42
ILUSTRACIÓN 30: MÓDULO COLA PRETALADRADO2	43
ILUSTRACIÓN 31: MÓDULO COLA PRETALADRADO1	44
ILUSTRACIÓN 32: MÓDULO TIEMPO DE PROCESO TOTAL 1	45
ILUSTRACIÓN 33: MÓDULO TIEMPO DE PROCESO TOTAL 2	45
ILUSTRACIÓN 34: MÓDULO DE SALIDA	46
ILUSTRACIÓN 35: RESULTADOS SIMULACIÓN 1.....	46
ILUSTRACIÓN 36: RESULTADOS SIMULACIÓN 2.....	47
ILUSTRACIÓN 37: ZOOM CUADRO ILUSTRACIÓN 32	47
ILUSTRACIÓN 38: TABLA DE TIEMPOS MEDIOS PIEZAS EN EL PROCESO (20 SEGUNDOS)	47
ILUSTRACIÓN 39: TABLA DE TIEMPOS MEDIOS PIEZAS EN EL PROCESO (15 SEGUNDOS)	48
ILUSTRACIÓN 40: VISTA GENERAL CÉLULA FLEXIBLE	49
ILUSTRACIÓN 41: INSTALACIONES LABORATORIO 1	49
ILUSTRACIÓN 42: INSTALACIONES LABORATORIO 2	50
ILUSTRACIÓN 43: BANDEJA	50
ILUSTRACIÓN 44: PLATAFORMA DEL LABORATORIO	51
ILUSTRACIÓN 45: MOTORES CINTA TRANSPORTADORA LABORATORIO.....	51

ILUSTRACIÓN 46: DIAGRAMA DE FLUJO DEL PROCESO 1	52
ILUSTRACIÓN 47: DIAGRAMA DE FLUJO DEL PROCESO 2	53
ILUSTRACIÓN 48: DIAGRAMA DE FLUJO DEL PROCESO 3	54
ILUSTRACIÓN 49: CONFIGURACIÓN DEL RACK O BASTIDOR EN UNITY PRO XL	55
ILUSTRACIÓN 50: PANTALLA PRINCIPAL UNITY PRO XL	56
ILUSTRACIÓN 51: INSTALACIONES LABORATORIO 2	57
ILUSTRACIÓN 52: CAPTURA SECCIÓN GESTIÓN_PRIORIDAD UNITY PRO XL.....	58
ILUSTRACIÓN 53: INSTALACIONES LABORATORIO 2	59
ILUSTRACIÓN 54: CAPTURA SECCIÓN PRIORITY_PT12 UNITY PRO XL	60
ILUSTRACIÓN 55: CAPTURA SECCIÓN RIGHT_BAND2 UNITY PRO XL	61
ILUSTRACIÓN 56: PRESUPUESTO	65

Resumen

Este trabajo se enmarca dentro del área temática de la automatización industrial y, concretamente, en el estudio de las etapas de automatización de un proceso industrial y sus implicaciones en la gestión de la producción.

Así, en este trabajo, partiendo de unas instalaciones industriales, se afronta en primer lugar el estudio de los diferentes escenarios y soluciones posibles para una optimización del proceso productivo propuesto mediante el uso de una herramienta profesional de para la simulación de procesos como es Arena Simulation. El análisis de los comportamientos del proceso, en términos de rendimiento ante diferentes variaciones, resulta la identificación de una solución robusta y eficaz.

En segundo lugar, la implementación de la configuración del proceso industrial se lleva a la programación en la celda industrial disponible afrontando la configuración y programación de los controladores lógicos programables mediante una herramienta profesional de trabajo con equipos de automatización, Unity Pro XL.

El proyecto finaliza con la validación del funcionamiento del proceso programado en entorno real y la validación parcial de su rendimiento todo en relación a las expectativas del diseño inicial.

1. Introducció

1.1 Objectiu

El principal objectiu del projecte es realitzar el estudi de les diferents etapes de automatització de un procés industrial considerant aspectes de operació i producció. Se consideren, per lo tanto, característiques de la organització del procés industrial mitjançant la simulació i anàlisi de la producció, i característiques de disseny de un automatisme programat mitjançant el desenvolupament i validació experimental sobre una cel·la industrial de la funcionalitat i el rendiment esperat.

1.2 Requisits

El projecte parteix d'un conjunt de requisits agrupats en tècnics i funcionals/operatius.

1. Els requisits tècnics són:

- Entorn de programació de PLCs de Schneider, Unity XL.
- Cèl·lula de treball, PLCs, actuadors i sensors basats en els components disponibles en el laboratori.

2. Per lo que respecta a els requisits funcionals i operatius:

- Operació en cicle continuu del procés industrial a dissenyar.
- Nombre de processos a considerar en funció de les línies que existeixen físicament en el laboratori.
- Monitorització de variables de rendiment, com els temps de producció.

1.3 Antecedents

El nou context de la indústria 4.0 o quarta revolució industrial té com un dels seus objectius la millora de processos a través de la tecnologia per reduir costos i temps productius. De esta manera, la monitorització d'indicadors clau de rendiment, així com el anàlisi de

la gestión de pedidos con el objetivo de optimizar los recursos disponibles representa un aspecto clave en las nuevas líneas de producción industrial.

La célula flexible son un grupo de máquinas-herramientas enlazadas entre sí mediante un sistema de transporte de piezas común y un sistema de control centralizado. Es un sistema de fabricación flexible (FMS).

Estos FMS ayudan a implementar la filosofía “Just-in-time”, que significa fabricar únicamente lo estrictamente necesario, cuando es necesario y solamente las cantidades imprescindibles. El objetivo es reducir stocks al mínimo necesario. [1]

El interés por el proyecto surge debido a la creciente tendencia de la producción flexible, es decir, la consideración de procesos productivos capaces de responder de manera rápida y eficaz a una producción por lotes como respuesta a una demanda variable del mercado. Por lo tanto, se hace necesario optimizar los procesos productivos para que puedan gestionar demandas productivas cambiantes o incluso que coexistan diferentes etapas de manipulación sobre el material de entrada a lo largo del proceso productivo que tengan lugar en función del pedido y producto final esperado.

Personalmente, y después de estar cursando las prácticas como becario en una empresa que se dedica a realizar inspecciones técnicas y compartir experiencia con los compañeros, me he dado cuenta de la enorme importancia y trascendencia de esta disciplina para un ingeniero que quiere introducirse en el diseño, programación y gestión de procesos industriales. Es por eso que el hecho de formarme en el ámbito de los autómatas programables supone un reto y motivación como profesional, además de que considero necesario adaptar mis conocimientos a este nuevo contexto surgido. A esto se suma la oportunidad de trabajar con la célula de automatización disponible en las instalaciones de la Universidad.

1.4 Alcance del estudio

El alcance del proyecto incluye:

- Definición del proceso industrial en base a las instalaciones reales disponibles en el laboratorio.
- Modelado del proceso productivo y análisis de operación.
- Selección y estudio de la solución óptima en función de tiempos, colas y rendimientos de producción.
- Diseño y programación del automatismo sobre autómatas lógicos programables.

De la misma manera, el proyecto no incluye aspectos de definición, diseño e implementación de elementos actuadores, sensores o lógicos relacionados con la implementación posterior del automatismo programado.

2. Descripción del proceso

El proceso considerado para llevar a cabo este proyecto simula una línea de producción donde se mecanizan piezas metálicas. Consiste en que al final del proceso se obtengan únicamente piezas completas, que han sido agujereadas y posteriormente roscadas.

Así, a la entrada de la línea de mecanizado pueden llegar piezas metálicas de tres tipos:

1. Piezas finalizadas (con agujero y rosca)
2. Piezas solamente agujereadas (falta rosca)
3. Piezas sin agujerear (falta agujero y rosca)

La cantidad de unidades de piezas que se incorpora al proceso se ha determinado después de realizar varias pruebas. Al ir aumentando el número de piezas y variando el tiempo entre entrada de piezas al sistema hemos observado que el sistema funcionaba correctamente entrando un lote de 500 piezas, introduciendo una unidad cada 20 segundos, hasta la finalización del lote.

En la línea de mecanizado, las piezas se desplazan a través de una cinta transportadora que presenta dos bifurcaciones a lo largo de su recorrido. En la primera bifurcación únicamente deben desviarse del cauce principal del proceso aquellas piezas que no han sido agujereadas, mientras que en la segunda bifurcación deben desviarse aquellas que tienen agujero pero no rosca.

Cada bifurcación permite hacer llegar las piezas a una zona de trabajo donde se le realiza la operación pertinente para que la pieza pueda avanzar en el proceso: en primer lugar el agujereado y en segundo lugar el roscado.

La *ilustración 1* muestra la representación del proceso definido, donde aparece la línea principal y las bifurcaciones que deben seguir las piezas metálicas con las distintas estaciones o zonas de trabajo (agujereado y roscado).

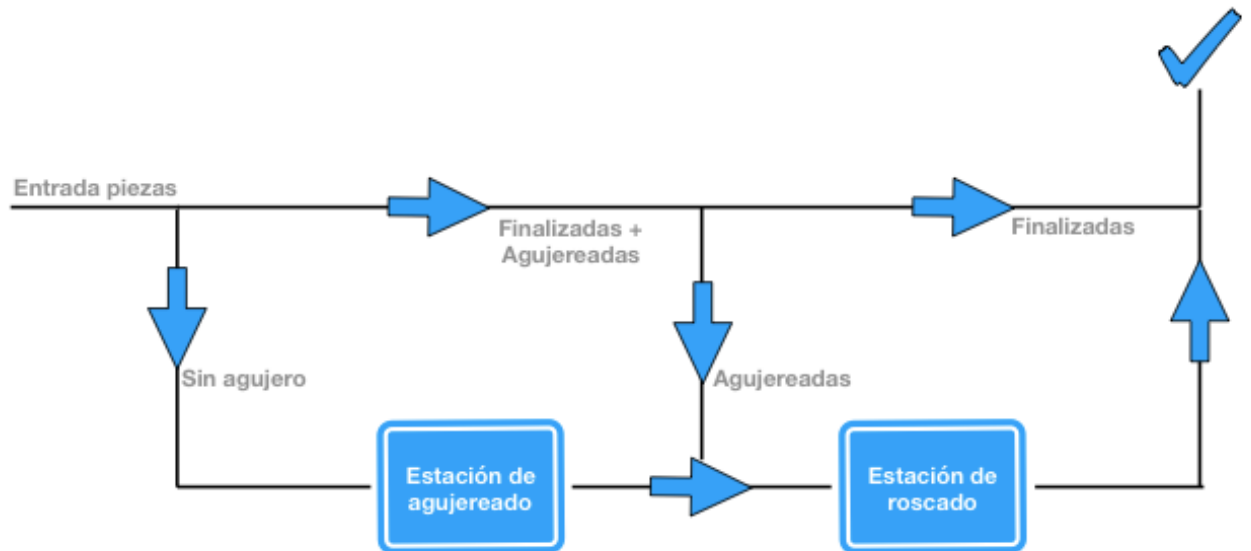


Ilustración 1: Esquema de proceso

3. Marco teórico software de simulación de procesos industriales

Existen simuladores especialmente dedicados al ámbito industrial, que tienen como objetivo mejorar e incrementar la eficiencia durante el diseño de los procesos de manufactura en estas industrias, y que simulan múltiples aspectos de los diferentes procesos antes de implantarlos físicamente. Estas simulaciones permiten obtener una serie de resultados para analizar y saber cuál sería el comportamiento de una futura realización de los mismos que se adapte mejor a las especificaciones de operación, producción y logística principalmente.

Algunos de estos programas son potentes herramientas de cálculo, con grandes bancos de datos que contienen las propiedades físicas de múltiples compuestos y sustancias químicas, modelos termodinámicos, cálculos de equipos, análisis de costes, estados y condiciones de operación entre otros, que le dan a estos entornos de simulación una gran profundidad de diseño y gran adaptabilidad al proceso y a las necesidades de análisis.

Aunque existe una amplia gama de aproximaciones a la simulación de procesos, tres de los entornos software más utilizados profesionalmente son:

→ Aspen HYSYS

Es un programa interactivo que está enfocado a la ingeniería de procesos y a la simulación, muy adecuado, sobretodo, en procesos químicos con un gran uso en la simulación de plantas petroquímicas. La interfaz es muy simple e intuitiva para el usuario, permitiendo el empleo de operadores lógicos y herramientas que facilitan la simulación de los procesos. Es un simulador bidireccional, ya que el flujo de información va en dos direcciones (hacia delante y hacia atrás), permitiendo calcular las condiciones de una corriente de entrada a una operación a partir de las correspondientes a la corriente de salida sin necesidad de cálculos iterativos. Además, posee

un entorno de simulación modular tanto para estado estacionario como para régimen dinámico.

→ CHEMCAD

Es un paquete con varios módulos que abarca cálculo y diseño de intercambiadores de calor (CC-THERM), simulación de destilaciones dinámicas (CC-DCOLUMN), simulación de reactores por lotes (CC-ReACS), simulación de destilaciones por lotes (CC-BATCH), simulación de redes de tuberías (CC-SAFETY NET). Este sistema es muy utilizado para el diseño, operación y mantenimiento de procesos químicos en una gran variedad de industrias incluyendo la explotación de petróleo y gas, farmacéuticos, biocombustibles y procesos químicos industriales en general. Este software se utiliza como una herramienta para mejorar la productividad, maximizar la rentabilidad de las operaciones por el diseño más eficiente de nuevos procesos y equipos, y por la reducción de costes e inversiones de capital por la optimización y solución de los cuellos de botella existentes en los procesos y en los equipos. [2]

→ Arena Simulation

Es un software de simulación de eventos discretos para la optimización de procesos complejos, que ofrece un mejor entendimiento de las cualidades de un sistema.

El modelado de eventos discretos es el proceso de representar el comportamiento de un sistema complejo como una serie de eventos bien definidos y ordenados en el tiempo. Esto permite analizar rápidamente y de forma eficaz el comportamiento de un proceso o sistema a lo largo del tiempo.

Dado un proceso, el software ARENA se puede utilizar para multitud de finalidades:

- Puede generar diferentes escenarios para buscar la solución a un problema sin una causa clara.

- Permite encontrar el mejor escenario minimizando el riesgo de una futura inversión.
- Identifica los cuellos de botella que tiene nuestro sistema o localiza cuellos sobredimensionados que son totalmente innecesarios.
- Permite dar visibilidad del efecto que tendría un cambio en nuestro proceso, además de darnos la capacidad de explorar nuevos procedimientos o escenarios sin tener que interrumpir el proceso actual .

También es importante destacar las ventajas que nos ofrece este programa para sacar el máximo rendimiento a un estudio:

- Bloques de construcción predefinidos para modelar el proceso sin necesidad de programación.
- Amplia gama de opciones de distribuciones estadísticas para modelar la variabilidad del proceso de forma precisa.
- Métricas de rendimiento, análisis estadístico y generación de informes.
- Capacidades realistas de animación en 2D y 3D.

Más allá de las muchas ventajas de este programa, este es el software utilizado para la simulación de nuestro proceso por planteamiento de la co-tutora del trabajo. La versión gratuita de este es más que suficiente para la simulación de nuestro proceso.

[3]

Mediante la siguiente tabla, con puntuaciones del 1 (peor) a 5 (mejor), se valoran las diferentes opciones :

	Aspen HYSYS	Chemcad	Arena Simulation
Facilidad de uso	5	5	5
Características y funcionalidad	3	4	4
Relación calidad-precio	2	3	3
Calidad versión Beta	3	1	5
TOTAL	13	13	17

Ilustración 2: Comparativa de softwares de simulación de procesos industriales

A raíz de los valores obtenidos en la tabla, el software escogido es Arena Simulation.

4. Marco teórico automatización de procesos industriales

La automatización de procesos industriales se refiere a todo aquello consistente en el uso de un sistema computarizado y electromecánico para el control de maquinarias o procesos industriales. Se basa en la capacidad para controlar la información necesaria en el proceso de producción. [4]

Los principales objetivos de la automatización de un proceso son mejorar la productividad, los estándares de calidad, la seguridad y obtener una alta flexibilidad y precisión de información. El tema de la seguridad es muy importante, ya que permite liberar a personas de la realización de ciertas tareas perjudiciales para la salud o con determinados niveles de peligrosidad. [5][6]

Para llevar a cabo esta automatización debemos conseguir una perfecta sincronización de los componentes que integran nuestro proceso. Para ello se utiliza un PLC o autómatas programables, el cual se encarga de tratar la información que recibe de unas entradas (normalmente pulsadores, sensores o selectores) y decidir que actuadores (por ejemplo válvulas, pistones o electroválvulas) o que parte de nuestro proceso debe activarse para que se cumpla o realice aquello que hemos programado previamente.

Por lo tanto los principales bloques que conforman el proceso son:

- PLC o autómatas programables
- Entradas o inputs
- Salidas o outputs

4.1 PLC o autómata programable

<https://www.ingmecafenix.com/automatizacion/que-es-un-plc/>

<http://automantenimiento.net/electricidad/partes-de-un-plc/>

Un PLC es básicamente una computadora que procesa todas las entradas o inputs, las cuales generalmente proceden de sensores, pulsadores o selectores. Para posteriormente controlar a los diferentes actuadores, como pistones o válvulas, y así controlar el proceso industrial de forma automática.

Según las características existen tres tipos de PLCs:

- **Nano:** son los más sencillos, con un número reducido de entradas y salidas.



Ilustración 3: PLC Nano

- **Compacto:** todos sus componentes se integran en un módulo principal. En este caso, aunque su tamaño es mayor que los nano, el número de entradas y salidas es limitado, ya que no puede expandir sus modelos.



Ilustración 4: PLC Compacto

→ **Modular:** tiene la ventaja que pueden ser configurados en función de las necesidades existentes, ya que cada módulo lo encontramos por separado, para posteriormente construir el PLC deseado.



Ilustración 5: PLC Modular

También debemos diferenciar los elementos básicos que forman un PLC:

→ Fuente de alimentación

Su función es proporcionar corriente continua a la CPU o procesador y demás módulos que conforman el sistema.

→ CPU o procesador

Es la parte inteligente del sistema. Controla la secuencia de ejecución del programa, interpretando las instrucciones de este. Para ello:

- Consulta el estado de las entradas y lo guarda en la zona de memoria imagen de las entradas, para que el programa pueda acceder a estos.
- Coordina la comunicación entre los diferentes componentes.
- Controla que el tiempo de ejecución no sea excesivo y que no haya errores ni de hardware ni software.
- Realiza operaciones lógicas e incluso aritméticas, para acabar ordenando la activación de las salidas correspondientes.

→ Módulos de entradas y salidas

Permiten realizar una conexión física entre la CPU y el sistema a controlar (diálogo hombre-máquina).

A través del módulo de entrada se recoge y se manda el estado de sensores, pulsadores o selectores, para que el PLC pueda procesar estos datos. Mientras que mediante el módulo de salida, una vez que el PLC ha recibido y procesado los datos en base a su programa, envía las señales correspondientes a los actuadores para controlar el proceso.

→ Bus del sistema

https://es.wikipedia.org/wiki/Bus_de_campo

Es un sistema de transmisión de información cuyo objetivo es sustituir las conexiones punto a punto, estandarizando y simplificando muy notablemente la instalación de máquinas y equipamientos industriales. Son redes digitales, bidireccionales y multipunto. [9]

Los más extendidos en el ámbito industrial son los buses CAN, Profibus y ASI.

→ Puerto de conexiones y comunicaciones

Permiten el intercambio de datos entre el autómata y un terminal de programación específico (como un ordenador).

Las conexiones autómata-ordenador más comunes son mediante Ethernet o usb.

→ “Rack” o bastidor

Es la base sobre la que se montan los módulos anteriormente comentados: fuente de alimentación, CPU, módulo de entradas y salidas (que pueden ser digitales o físicas) y módulo de comunicaciones. A través de este se transmite la información entre la CPU y los módulos, a la vez que se alimentan. [7][8]

El PLC del laboratorio es el modelo M340 de la marca Schneider Electric. Tiene un rack con 8 slots o espacios.

4.2 Ciclo de SCAN

Se conoce como ciclo de SCAN a una secuencia de operaciones que el autómata realiza una y otra vez, de forma repetitiva, cuando entra en modo RUN.

El conjunto de operaciones que el autómata realiza de forma cíclica podemos dividirlos en:

1. **Lectura estados entrada:** comprueba cada una de las entradas, para ver si están activas o no, y esta información la guarda en su memoria para utilizarla en la siguiente etapa.
2. **Ejecución del programa de control:** se ejecutan de forma secuencial las líneas del programa, escribiendo el resultado lógico en las salidas de la imagen del proceso.
3. **Actualización del estado de las salidas:** el resultado, almacenado en las imágenes del proceso, se escribe en las salidas físicas del autómata.

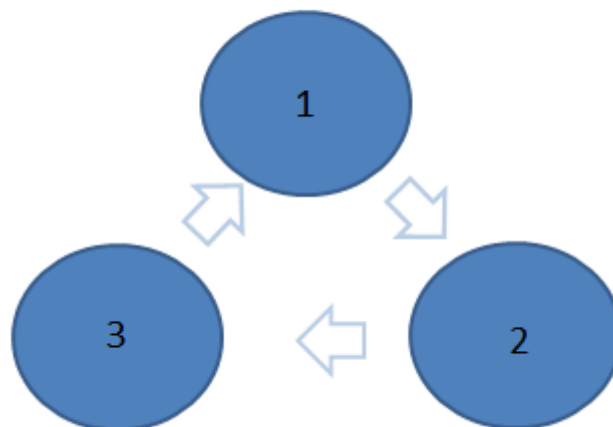


Ilustración 6: Ciclo de SCAN

La duración de este proceso suele oscilar entre 1 y 100 milisegundos, dependiendo del procesador. [10][11]

4.3 Detectores y actuadores

- **Detectores inductivos**

[12] <https://es.slideshare.net/josueacerov/sensores-inductivos-y-plc>

Sirven para detectar materiales férricos sin necesidad de contacto físico, se basa en la variación de los campos electromagnéticos.

El sensor tiene una bobina en su interior, que cuando circula corriente crea un campo magnético. Si un metal conductor se mueve dentro de un campo magnético se generan corrientes eléctricas conocidas como corrientes de Foucault. Estas corrientes a su vez generan un campo magnético opuesto al generado por la bobina del detector. Esto provoca una disminución de la amplitud de onda del campo magnético resultante. Cuando esta amplitud de onda se reduce hasta un determinado valor, el sensor entra en etapa de detección. [12]



Ilustración 7: Detector inductivo con retenedor

Forma de onda de la oscilación

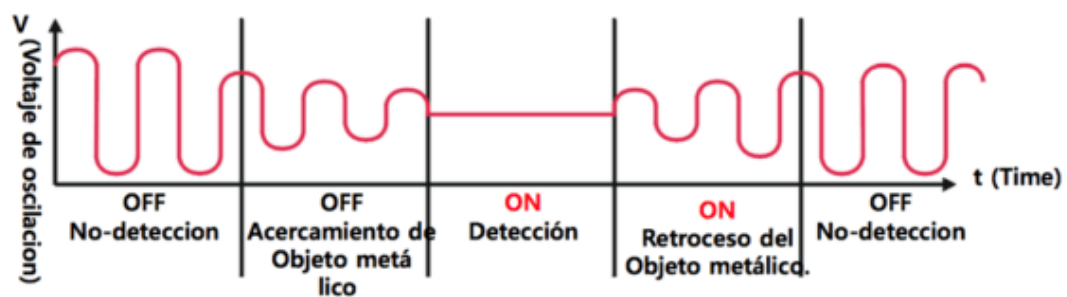


Ilustración 8: Oscilación onda detector inductivo

- **Electroválvulas neumáticas**

Su función es distribuir el aire a presión a todos los elementos neumáticos. Estos elementos neumáticos tienen un solenoide que, cuando se dan las condiciones especificadas en la programación de cada elemento, recibe una tensión que lo activa y permite el paso de este aire a presión. Cuando estas condiciones cambian deja de recibir tensión, bloqueando el paso del aire a presión. Hay dos en nuestra línea.



Ilustración 9: Electroválvula neumática

4.4 Programación del PLC

4.4.1 Contexto histórico

El desarrollo inicial de los PLCs surgió debido a la demanda de los fabricantes de automóviles para reducir el enorme coste que significaba el reemplazo de un sistema basado en relés. Fue en 1968 cuando la compañía americana Bedford Associates creó los primeros autómatas programables.

A principios de los años 70 estos ya incorporaban el microprocesador. En el 1973 aparecieron los PLCs con capacidad de comunicación (Modbus de MODICON). De este modo, los PLCs eran capaces de intercambiar información entre ellos y de situarse lejos de los procesadores y de los objetos que iban a controlar. También se añadieron más funcionalidades como manipulación de datos, cálculos matemáticos, elementos de

comunicación hombre-máquina, etc. A mediados de los años 70 apareció la tecnología PLC.

En los años 80 se produjo un intento por estandarizar las comunicaciones con el protocolo MAP (Manufacturing Automation Protocol) por parte de General Motors. También se consiguió una reducción de las dimensiones de los PLC's y se pasó a programar con un lenguaje simbólico a través de ordenadores personales en lugar de los clásicos terminales de programación. Las principales mejoras en cuanto a prestaciones que se llevaron a cabo en esta década están relacionadas con: velocidad de respuesta, reducción de dimensiones, concentración del número de entradas/salidas en los respectivos módulos, desarrollo de módulos de control continuo, PID, servo controladores, control inteligente y fuzzy (borroso).

En la década de los 90 se produjo una reducción gradual del número de protocolos nuevos. El último estándar, IEC 1131-3, trata de unificar el sistema de programación de todos los PLC en un único estándar internacional.

Hoy en día se dispone de PLCs que pueden ser programados en diagramas de bloques, listas de instrucciones, Ladder o texto estructurado al mismo tiempo. Además, la tendencia actual es la de dotar al PLC de funciones específicas de control y canales de comunicación para que pueda conectarse entre sí y con ordenadores en red, creando así una red de autómatas.
[13][14]

4.4.2 Lenguajes de programación

Los lenguajes de programación se pueden agrupar según si utilicen símbolos o texto para declarar las instrucciones de control en:

- **Tipo gráfico:** estos programas utilizan símbolos para declarar las instrucciones de control. Estos son Function Block Diagram (FBD) y Ladder (LD).

→ **Tipo literal:** estos programas utilizan texto estructurado para declarar las instrucciones de control . Estos son Instruction List (IL) y Structured Text (ST).

En la realización del proyecto se ha utilizado mayoritariamente el lenguaje Ladder y una parte ha sido programada en texto estructurado.








4.4.2.1 Lenguaje Ladder

El lenguaje Ladder se basa en la representación de esquemas eléctricos de relés y contactores.

Consta de dos líneas verticales con tensión. Cada etapa o escalón se muestra como una línea horizontal. Estas líneas horizontales tienen contactos y bobinas. Para que una bobina pueda activarse deben estar cerrados todos los contactos de la línea para que pueda pasar la corriente.

Estos contactos de las líneas horizontales pueden estar asignados a variables de entrada del sistema, por ejemplo que un contacto que detecta transiciones positivas deja pasar la corriente cuando se acciona un pulsador, pero únicamente durante un ciclo de SCAN. Por otro lado existen contactos que se asocian a variables internas del proceso, que son las que se usan para por ejemplo realizar acciones secuenciales. Básicamente se utilizan con el fin de facilitarnos la programación. [15]

La tabla correspondiente a la ilustración 10 muestra los componentes principales de este lenguaje, según las normas NEMA (National Electrical Manufacturers Association) [16]:

Símbolo	Descripción
	Contacto normalmente abierto: el contacto se encuentra en circuito cerrado cuando su valor lógico es 1 mientras que se encuentra en circuito abierto cuando su valor lógico es 0.
	Contacto normalmente cerrado: el contacto se encuentra en circuito cerrado cuando su valor lógico es 0 mientras que se encuentra en circuito abierto cuando su valor lógico es 1.
	Contacto para detectar transiciones positivas: se activa en el momento que el estado lógico del contacto pasa de 0 a 1, pero únicamente sucede durante el ciclo de SCAN que se detecta la transición.
	Contacto para detectar transiciones negativas: se activa en el momento que el estado lógico del contacto pasa de 1 a 0, pero únicamente sucede durante el ciclo de SCAN que se detecta la transición.
	Bobina: cuando se active pondrá la bobina a nivel lógico 1.
	Bobina negada: cuando se active pondrá la bobina a nivel lógico 0.
	Bobina Set: cuando se active pondrá la bobina a nivel lógico 1 y aunque cambien las condiciones de entrada,




	se mantiene este nivel lógico.
	Bobina Reset: cuando se active pondrá la bobina a nivel lógico 0 y aunque cambien las condiciones de entrada, se mantiene este nivel lógico.
	Bobina para detectar transiciones positivas: cuando se active pondrá la bobina a nivel lógico 1 durante un ciclo de SCAN.
	Bobinas para detectar transiciones negativas: cuando se active pondrá la bobina a nivel lógico 0 y aunque cambien las condiciones de entrada, se mantiene este nivel lógico durante un ciclo de SCAN.

Ilustración 10: Tabla lenguaje Ladder

Una vez dentro del programa, en la parte superior de estos elementos se indica la entrada, salida o marca interna a la que corresponden, facilitando así su identificación.

Para facilitar la comprensión del uso de algunos componentes de la tabla se presenta un ejemplo de aplicación simple:

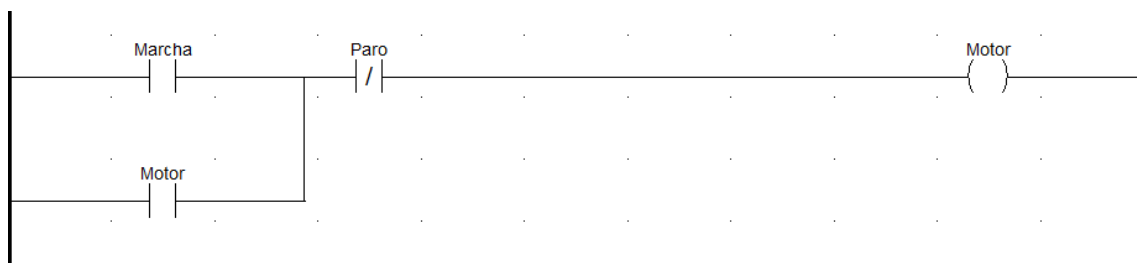


Ilustración 11: Ejemplo Ladder

En el siguiente esquema se observa que, mientras el pulsador de marcha no se active (que su valor lógico sea 1) no se activará el motor, ya que al ser un contacto abierto no deja pasar la corriente. El contacto abierto de la línea

inferior permite que el motor siga encendido hasta que se pulse el botón de paro, ya que si este contacto no estuviera presente el motor únicamente estaría encendido durante un ciclo de SCAN.

Este contacto abierto es comúnmente conocido como auto enclavamiento. El motor únicamente se apagará si se pulsa el botón de paro, ya que el contacto se abriría y dejaría de pasar la corriente.

Además de los elementos presentes en esta tabla, también podemos encontrar temporizadores y contadores. Estos elementos son realmente útiles y necesarios en la realización de nuestro programa, facilitando en gran parte nuestro trabajo debido a su fácil programación.

Existen tres tipos de temporizadores: los que permiten gestionar retrasos en la conexión (T_{ON}), los que permiten gestionar retrasos en la desconexión (T_{OFF}) y los que permiten realizar una pulsación de una duración determinada (T_P).

Por otro lado, existen tres tipos de contadores: los que cuentan de forma ascendente (CTU), los que cuentan de forma descendente (CTD) y contadores ascendentes/descendentes (CTUD).

4.4.2.2 Texto estructurado

Este lenguaje de programación se compone de una serie de instrucciones que se ejecutan de forma condicionada. Es muy similar al lenguaje C y, sobretodo, a Pascal.

Utiliza condiciones como IF, THEN o ELSE o también bucles secuenciales como WHILE o DO.

Un ejemplo simple de aplicación del lenguaje de texto estructurado se presenta a continuación:

```
IF (Contador1_ValorActual = X AND Condición1 = 0)
Acción1 := TRUE ;
ELSE
Acción1 := FALSE ;
END_IF ;
```

Ilustración 12: Ejemplo texto estructurado

Se observa que para el caso en que el valor actual de nuestro Contador1 sea una cantidad específica X y que Condición1 sea falsa, es decir, que no se de esta condición, entonces Acción1 se activará, si no se cumplen los requisitos iniciales Acción1 no se activará.

5. Diseño y optimización del proceso industrial

5.1 Metodología

La creación de un modelo de simulación de un proceso industrial que se ajuste al escenario real requiere de una serie secuencial de etapas:

- **Planteamiento del problema:** en esta etapa es necesario conocer el sistema real objetivo del análisis, para poder ver cual o cuales son los problemas que se necesitan solucionar. A partir del conocimiento del proceso y determinadas las variables de entrada y salida, se plantea un primer diagrama de flujo.
- **Recogida de datos:** conseguir todos los datos necesarios para poder reproducir fidedignamente el sistema real en la simulación.
- **Modelización del sistema:** proceder a construir el modelo. En el siguiente apartado se define con detalle esta etapa.
- **Verificación:** comprobar la consistencia interna entre el modelo lógico y el ordenador.
- **Validación:** comprobar si hay correspondencia entre el modelo de simulación y el sistema real.
- **Simulación:** en este punto se realiza la simulación de los diferentes escenarios posibles.
- **Análisis de resultados:** en esta última etapa se realiza un análisis de los datos obtenidos de la simulación, con el objetivo de entender el comportamiento del sistema y de esta forma poder dar solución a problemas planteados.

La ilustración 13 representa en esquema, las etapas que debe seguir el proceso de modelización para la simulación de un proceso industrial.

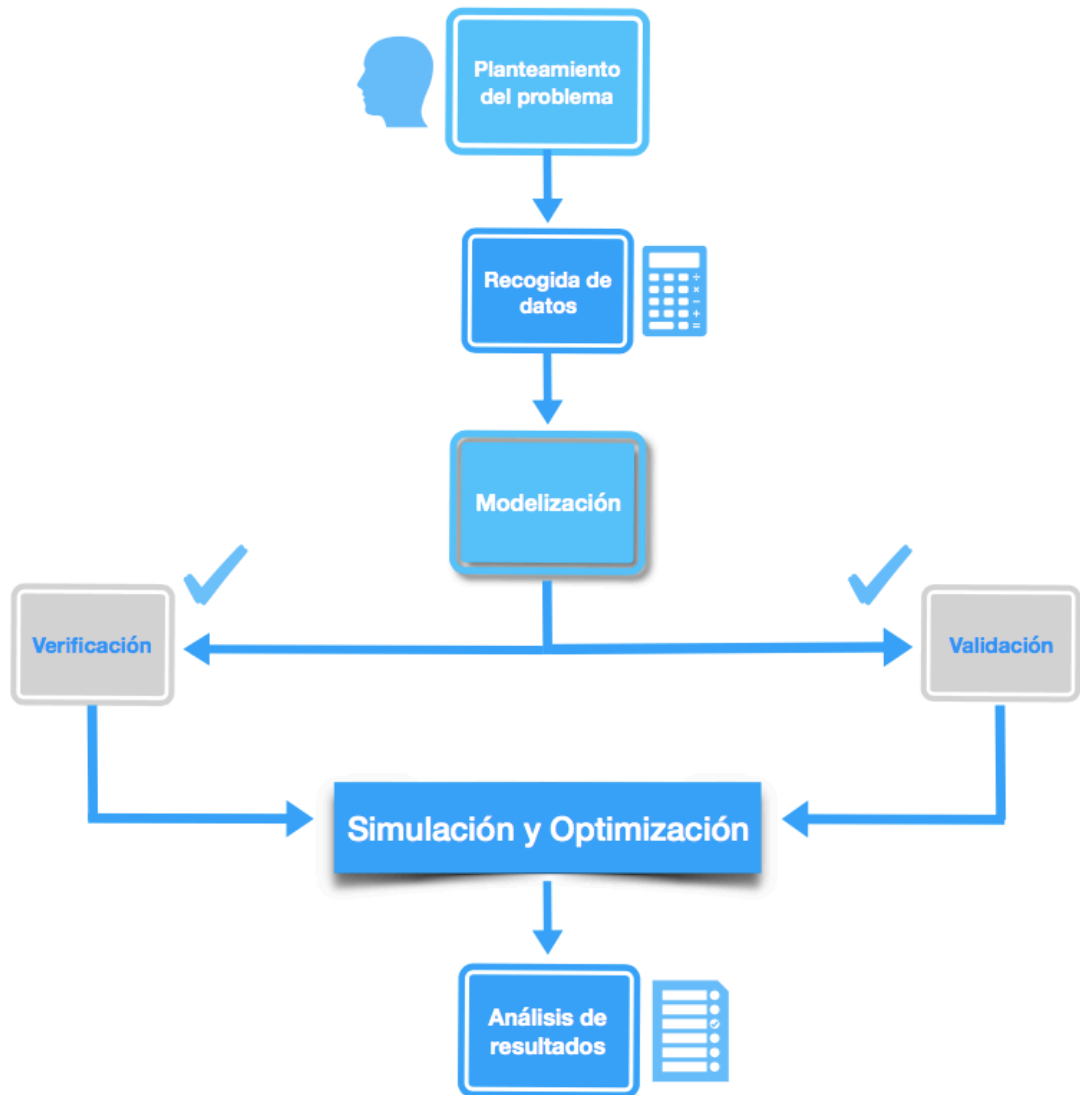


Ilustración 13: Esquema proceso de modelización para la simulación de un proceso industrial

5.2 Construcción y parametrización del modelo de simulación

De la descripción del proceso utilizado en el TFG, se generan siete escenarios diferentes según los siguientes criterios:

- Consecuencia de los diversos tratamientos de mecanizado de las piezas
- Gestión de la prioridad de movimiento en la línea del proceso.

La prioridad de movimiento se reflejará en el caso de que, se encuentren una pieza procedente de la estación de agujereado con otra pieza que ya presenta agujero, pero que en las dos piezas les falta el roscado.

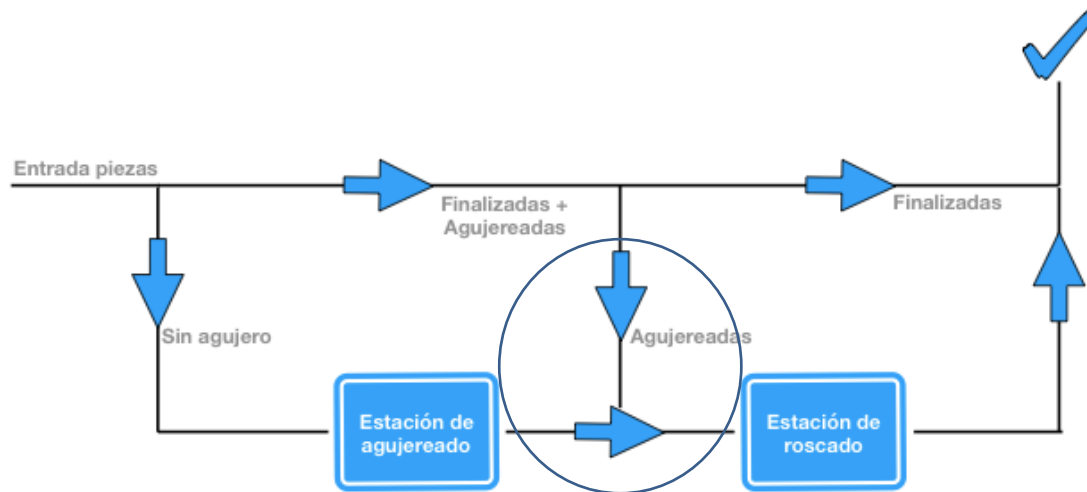


Ilustración 14: Esquema de proceso con punto gestión prioridad

Los diferentes escenarios son:

1. FIFO (Firs In First Out): la primera pieza que llega a la cola es la primera en acceder a la estación de roscado.
2. NANR (No Agujero No Rosca): tienen prioridad absoluta las piezas procedentes de la estación de agujereado sobre las que ya tenían agujereado pero les faltaba el roscado.
3. SANR (Si Agujero No Rosca): tienen prioridad absoluta las piezas que entran al proceso con agujereado pero sin roscado sobre las procedentes de la estación de agujereado.

La gestión de la prioridad en los diferentes casos se realiza comparando los tamaños de las colas que se generan a la salida de la estación de agujereado con las que se generan con las piezas que entran al proceso con agujereado pero sin roscado.

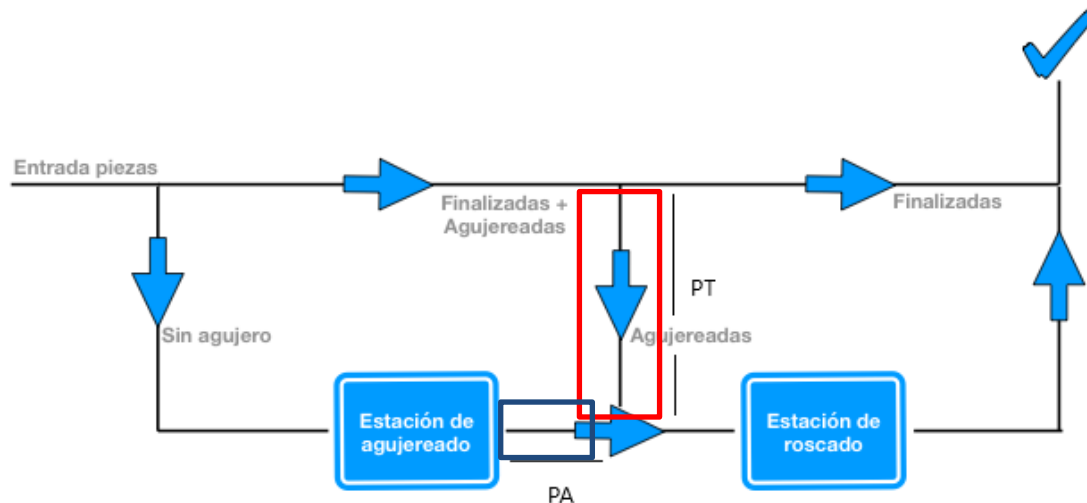


Ilustración 15: Esquema de proceso con colas PA y PT

En la ilustración 15 se observa las dos colas que se forman:

- PA (Post Agujereado), que se refiere a la cola que se genera en la zona inmediatamente posterior a la estación de agujereado, recuadrada en azul.
- PT (Pre Taladrado), que se refiere a la cola que se genera en la zona posterior a la bifurcación, recuadrada en rojo.

De esta situación surgen los 4 escenarios restantes. Aclarar que la primera prioridad es la condición que se debe cumplir para que avancen las piezas de la cola PA y la segunda prioridad es la condición para que avancen las piezas de la cola PT. Además, cuando se nombra PA o PT se hace referencia al tamaño de las colas que se producen en esa zona.

4. Prueba 1: $PA \leq PT / PT > PA$
5. Prueba 2: $PA \geq PT / PT < PA$
6. Prueba 3: $PA < PT / PT \geq PA$
7. Prueba 4: $PA > PT / PT \leq PA$

En este punto, se debe obtener una visión global de nuestro sistema. Aclarar que la configuración de los diferentes escenarios es bastante similar. Las diferencias entre ellos se van a explicar en la etapa correspondiente.

La ilustración 16 representa el proceso completo en que se basa la simulación.

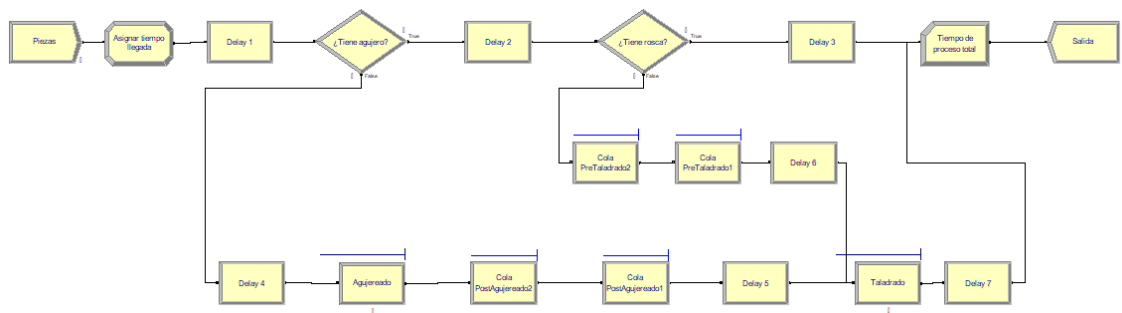
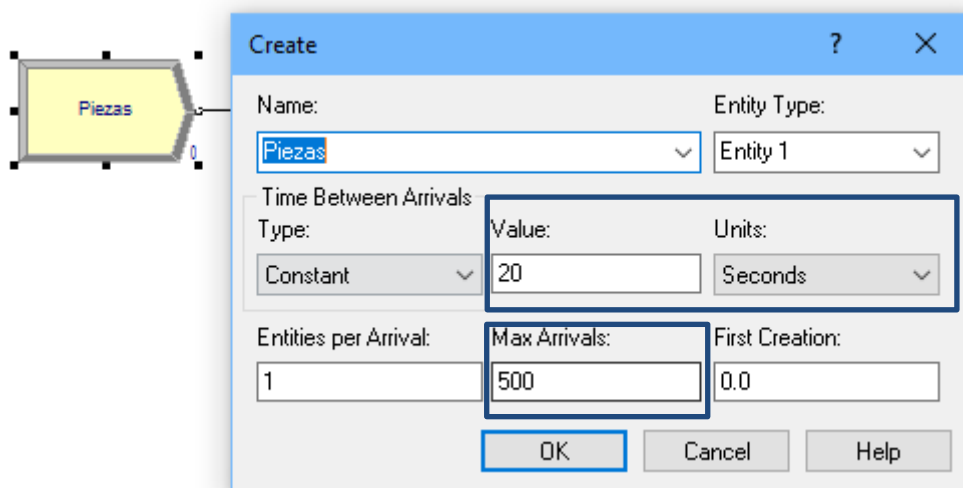


Ilustración 16: Proceso completo de la simulación en Arena

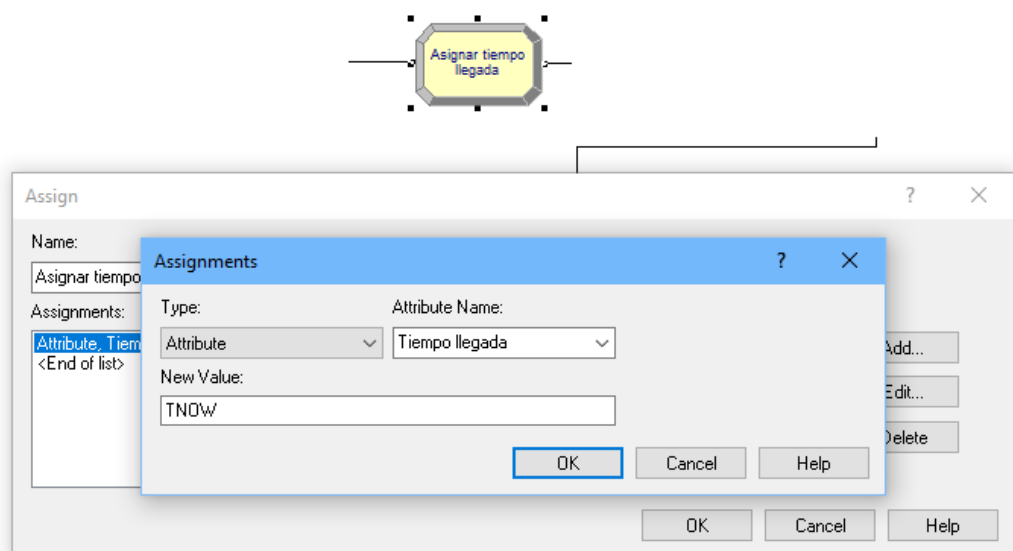
A partir de la visión global del proceso, se analizan los diferentes componentes (módulos) que lo forman con su configuración.

- 1. Creación de entidades:** este módulo *Create* llamado “Piezas” simula la llegada de las piezas. Como podemos observar llegarán 500 piezas, con un tiempo entre llegadas de 20 segundos.



Il·lustració 17: Mòdul de creació de entitats

- 2. Asignación de atributos:** este primer módulo del tipo *Assign* llamado “Asignar tiempo de llegada” se utiliza para asignar un atributo a cada entidad que pase por este módulo. Gracias a este atributo se asigna el valor del tiempo y nos permite saber el tiempo que esta entidad ha estado en nuestro sistema. Al asignar este valor del tiempo, introducimos como valor de atributo TNOW (Time Now), que en castellano significa tiempo ahora.



Il·lustració 18: Mòdul asignación de atributos 1

Este módulo de asignación de atributos también aparece en los escenarios NANR y SANR para asignar un valor a las piezas.

Las piezas por defecto tienen asignado el valor 0. Con este módulo, llamado “*Prioridad*”, se le asigna el valor de 1 para posteriormente en la operación de agujereado poder ordenar las piezas según el valor del atributo (mayor o menor), dando preferencia a las correspondientes para cada caso.

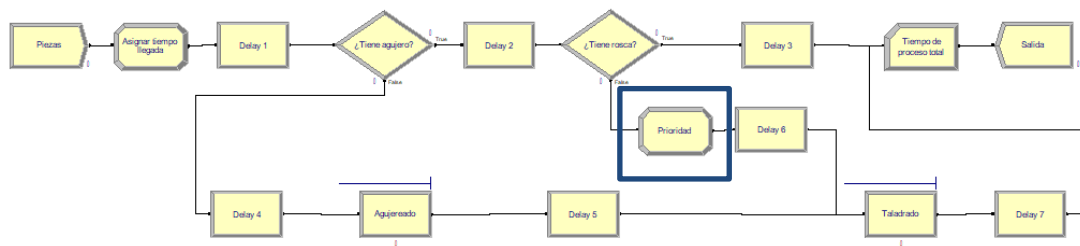


Ilustración 19: Módulo Prioridad

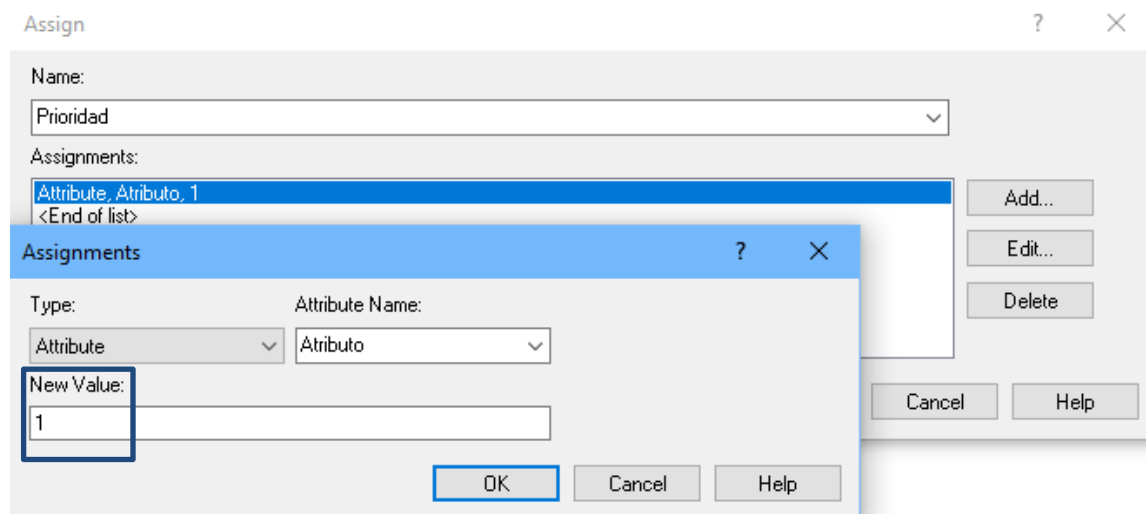


Ilustración 20: Módulo asignación de atributos 2

Tal como se ha definido, se asigna a las piezas que entran al proceso con agujero pero sin rosca el valor 1. Posteriormente en la operación de taladrado comparamos el valor de todas las piezas que llegan para poder ser ordenadas.

Si se entra dentro del módulo de Taladrado y se hace click en colas, se observa para el caso en que se prioriza las piezas que entran con

agujero sin rosca como se da preferencia a las piezas con mayor valor de atributo:

2	Taladrado.Queue	Highest Attribute Value	Attribute 1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
---	-----------------	-------------------------	-------------	--------------------------	-------------------------------------

Ilustración 21: Configuración cola Taladrado 1

Mientras que para el caso en que prioriza las unidades que entran sin agujero ni rosca, se da preferencia a las que tienen un menor valor de atributo:

2	Taladrado.Queue	Lowest Attribute Value	Attribute 1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
---	-----------------	------------------------	-------------	--------------------------	-------------------------------------

Ilustración 22: Configuración cola Taladrado 2

3. Demora: a lo largo del proceso se encuentran varios módulos del tipo *Delay*. En estos módulos, se añade a las piezas el tiempo en que se están desplazando por nuestro proceso.

En función del recorrido que realizan se encuentran con unos módulos *Delay* u otros, que le añaden distintos tiempos de desplazamiento. Cada tramo de nuestro proceso tiene su propia demora. Estos tiempos de desplazamiento han sido cronometrados en el laboratorio.

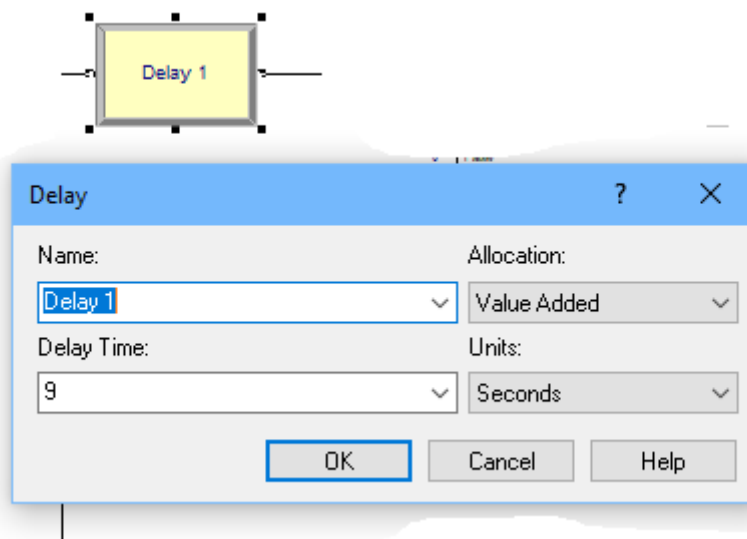


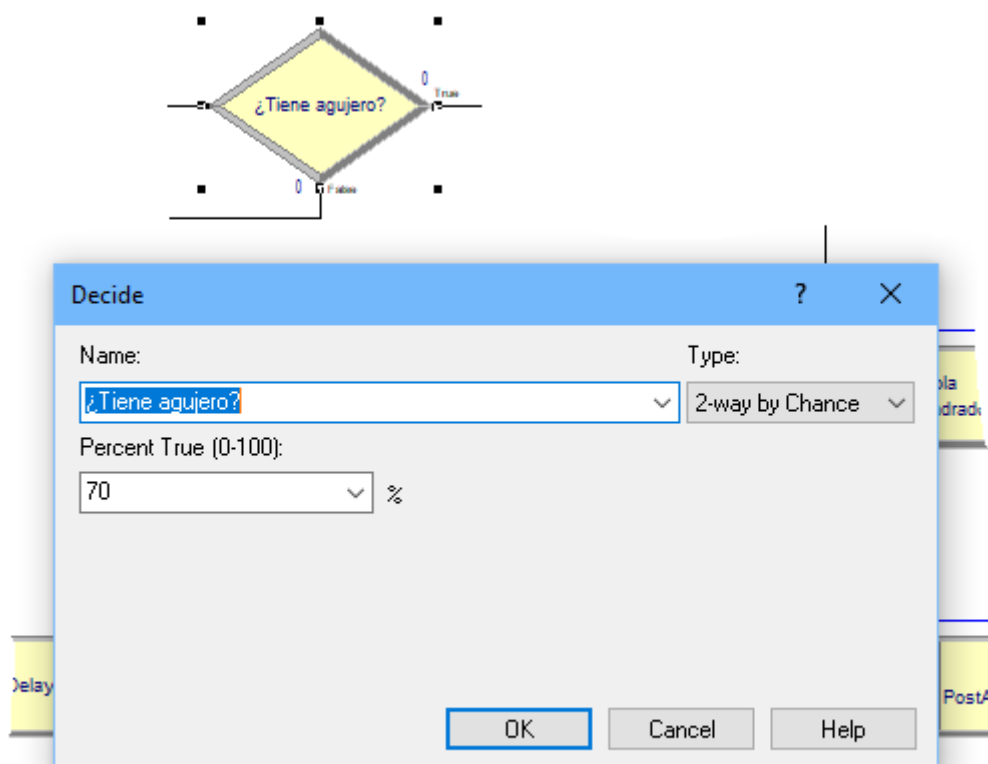
Ilustración 23: Módulo Demora

El tiempo que añade el “*Delay 1*” es de 9 segundos. En la siguiente tabla se pueden ver las diferentes demoras en los distintos tramos del proceso:

Nombre módulo	Valor módulo (segundos)
Delay 1	9
Delay 2	10
Delay 3	7
Delay 4	7
Delay 5	5
Delay 6	7
Delay 7	11

Il·lustració 24: Taula demoras

4. **Separación de piezas NANR del resto:** esta operación se realiza con un módulo *Decide*. Con este módulo se decide que un 70% de las piezas ya entran en el proceso con agujero realizado. Es decir, el 70% de las unidades avanza por el proceso, mientras que el 30% restante se dirigen a la operación de agujereado.



Il·lustració 25: Mòdul de separació de peces NANR del resto

5. **Separación de piezas SANR de correctas:** de la misma manera que en el punto anterior, se decide que el 70% de las piezas (49% del total) restantes están finalizadas y continúan avanzando. Por tanto el 30% de las piezas (21% del total) restantes se dirigen a la estación de taladrado.

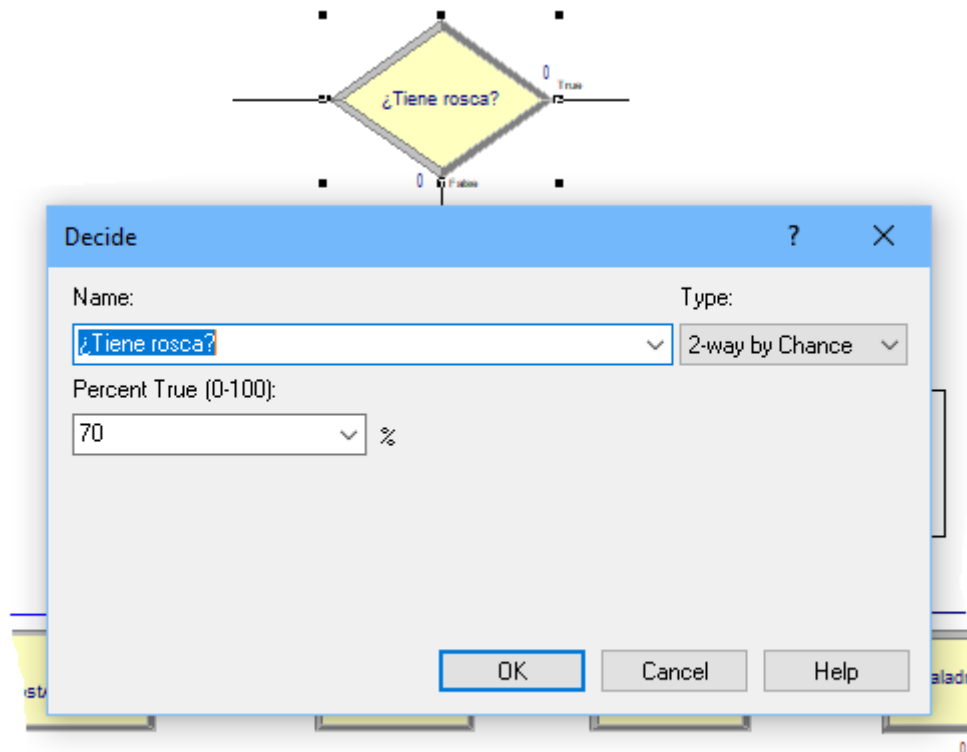
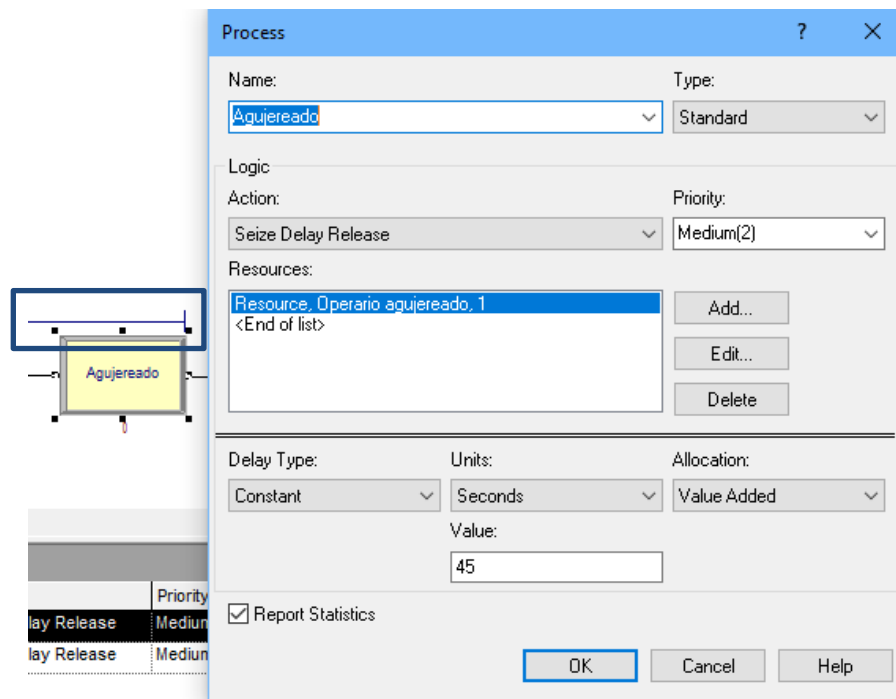


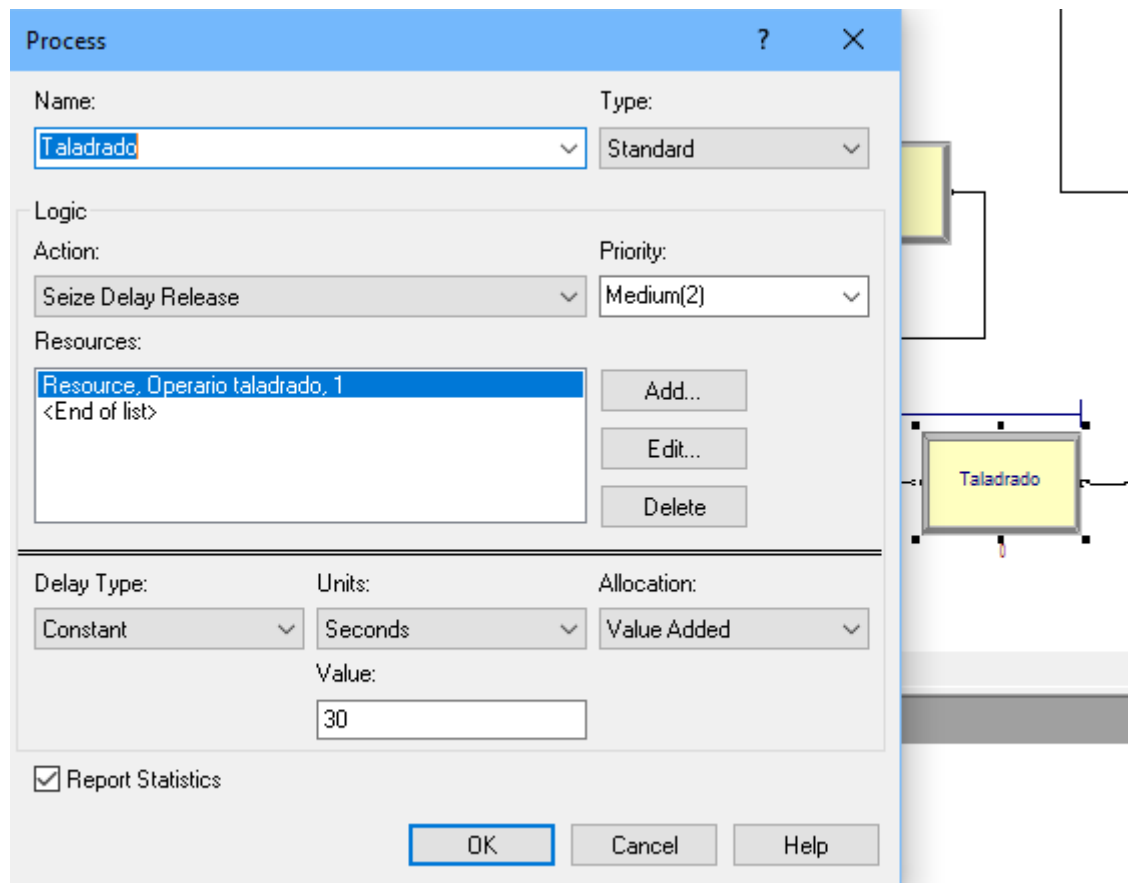
Ilustración 26: Módulo de separación de piezas SANR de correctas

- 6. Operación de agujereado:** con este módulo del tipo *Process* simulamos la operación de agujereado. Como se puede observar, se crea un recurso Seize-Delay-Release con el nombre de Agujereado. Esta operación tiene una duración fija de 45 segundos. En la parte superior de este módulo (encuadrado en azul) se puede observar la cola que se genera.



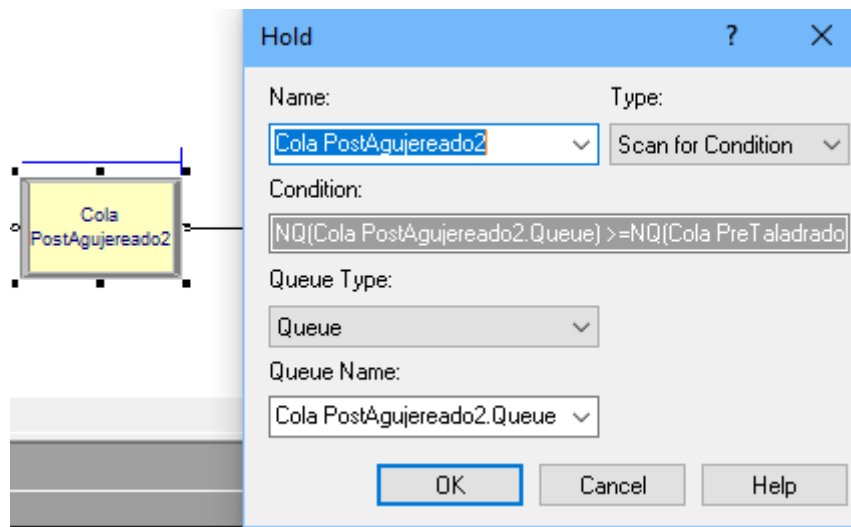
Il·lustració 27: Mòdul Agujereado

- 7. Operación de taladrado:** con esta operación conseguimos el roscado. De la misma manera que para el caso anterior, esta vez el recurso tiene el nombre de Taladrado, con una duración fija de 30 segundos. En este caso también podríamos observar la cola, pero en la operación de taladrado no se producirán colas como quedará claro en el punto número 9 de este mismo apartado 5.2. Construcción y parametrización del modelo de simulación.



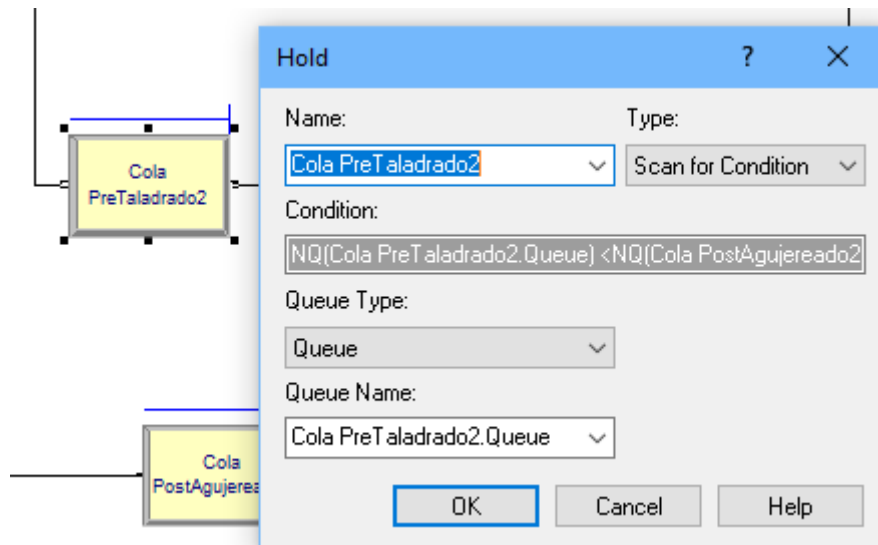
Il·lustració 28: Mòdul Taladrado

8. Cola PostAgujereado2 y PreTaladrado2: se utilizan módulos *Hold*. Con este módulo lo que se consigue es recrear una cola que está regida por una condición (explicadas al inicio de este mismo apartado 5.2 Construcción y parametrización del modelo de simulación). Para el caso en concreto de la Prueba2 las condiciones son las siguientes:



Il·lustració 29: Mòdul Cola PostAgujereado2

Para la “Cola PostAgujereado2”, avanzan piezas en el caso que el tamaño de su propia cola sea mayor o igual que el tamaño de la cola del módulo “Cola PreTaladrado2”.

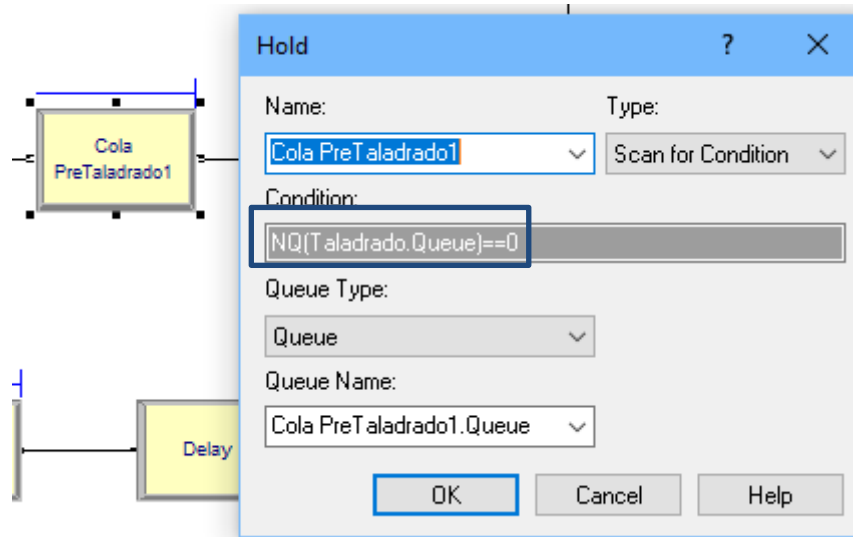


Il·lustració 30: Mòdul Cola PreTalladrado2

Mientras que para “Cola Pretalladrado2” avanzan piezas cuando el tamaño de su propia cola es menor que el tamaño de la cola del módulo “Cola PostAgujereado2”.

Para los diferentes escenarios únicamente variando los símbolos $<$, $=$ o $>$ se consiguen las diferentes combinaciones.

9. **Cola PostAgujereado1 y PreTaladrado1:** de nuevo se utilizan módulos *Hold* que permiten el avance de piezas cuando se da una condición determinada. La función de estos es muy concreta, ya que únicamente deben liberar una entidad de su cola cuando la cola del módulo de “*Taladrado*” sea cero. Ambos módulos tienen configuración idéntica, variando el nombre.



Il·lustració 31: Mòdul Cola PreTaladrado1

- 10. Registro tiempo llegada:** se hace mediante un módulo *Record*. Pretendemos registrar el valor del atributo previamente creado en el punto número 2 del apartado 5.2. Como lo que medimos es un intervalo de tiempo, se selecciona Time Interval como tipo de módulo. Le introducimos Tiempo de llegada como nombre al atributo y se define el Tiempo de proceso total como Tally name. Este último es el nombre con el que serán recogidos los resultados.

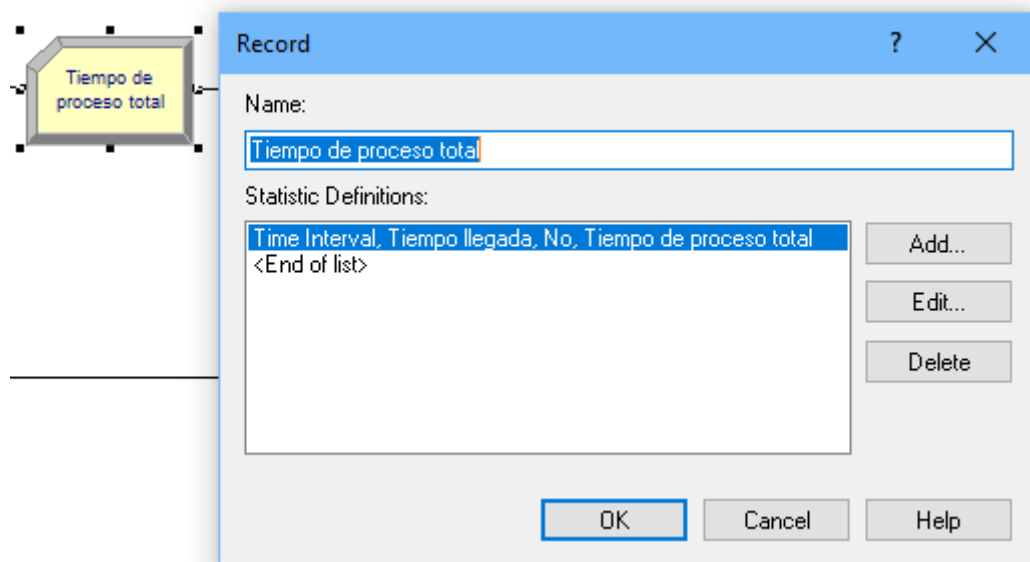


Ilustración 32: Módulo Tiempo de proceso total 1

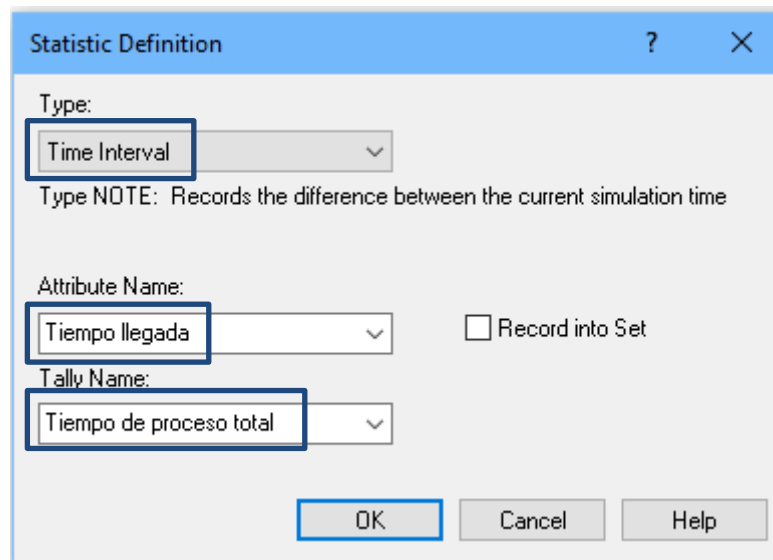


Ilustración 33: Módulo Tiempo de proceso total 2

- 11. Salida:** se representa con un módulo *Dispose*. Este módulo es el final de la simulación, donde desaparecen las entidades que han sido procesadas. Es de muy simple configuración, donde se debe introducir el nombre y seleccionar la opción Record Entity Statistics, para que se almacenen todas las estadísticas de las entidades.

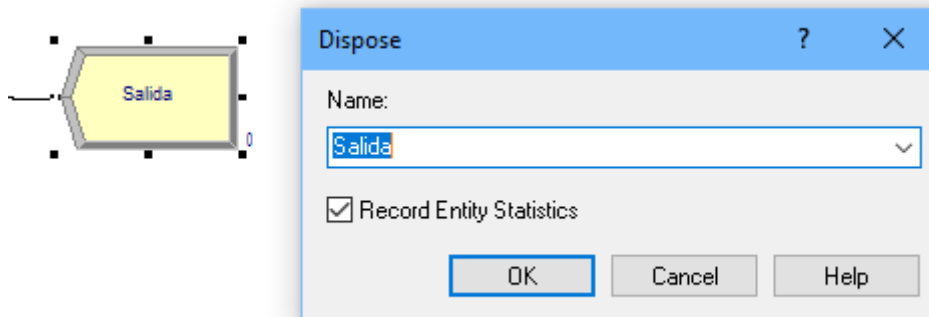


Ilustración 34: Módulo de salida

5.3 Definición de métricas

Con el objetivo de poder escoger el mejor escenario de las diferentes posibilidades existentes se deben simular todos y cada uno de los escenarios. El programa Arena ofrece los datos de la simulación en un desplegable que se puede descargar en formato PDF. Una vez finaliza la simulación el mismo programa pregunta si se desean obtener los resultados de la simulación.

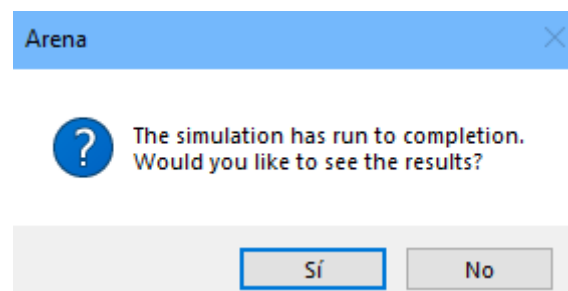
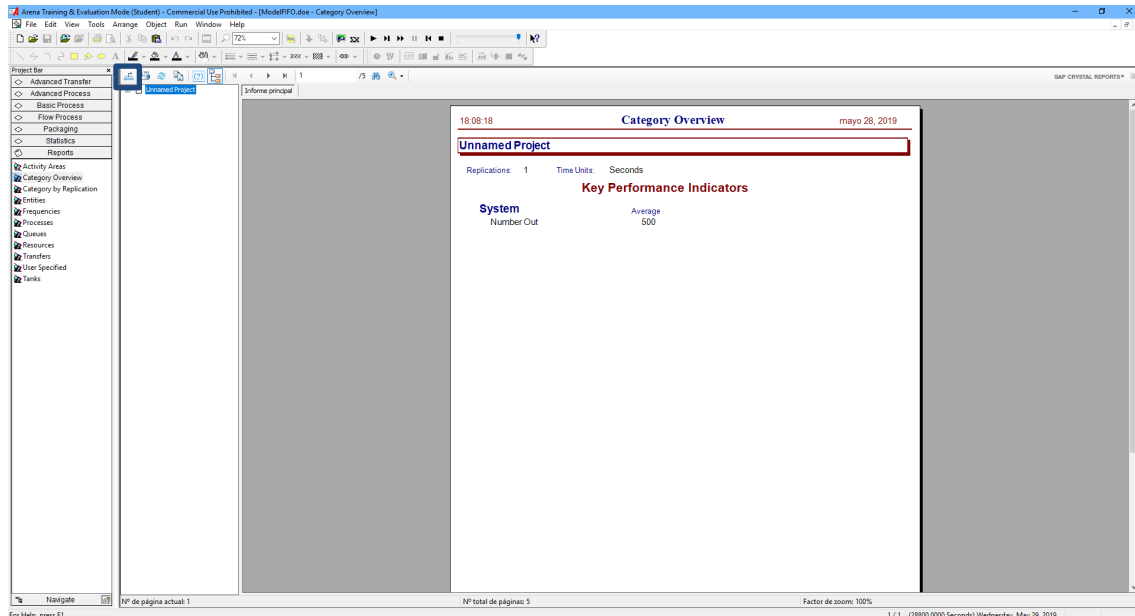


Ilustración 35: Resultados simulación 1

Si la respuesta es afirmativa aparecen los resultados en forma de desplegable en el mismo programa. Para poder descargarlos es tan simple como exportar los resultados en el formato deseado.



Il·lustració 36: Resultados simulación 2



Il·lustració 37: zoom cuadro ilustración 32

Todos los resultados de las diferentes simulaciones están en los Anexos.

Una vez realizadas las simulaciones que representan cada uno de los escenarios propuestos, se analizan los diferentes resultados. Se realiza una tabla con el tiempo medio en segundos que una unidad pieza permanece en el proceso en cada situación.

Este dato es posible obtenerlo gracias al módulo *Assign* que asignaba a todas las entidades el atributo “Asignar tiempo llegada” (explicado en el punto número 2 del apartado 5.2) y gracias al módulo *Record* (explicado en el punto número 10 del apartado 5.2) que registra el valor de este atributo de todas las entidades que pasan por el.

Nombre simulación	Tiempo medio de permanencia las piezas en el proceso si la llegada de piezas es cada 20 segundos
FIFO	77,06 segundos
Prioridad NANR	77,19 segundos
Prioridad SANR	77,19 segundos
Prueba 1	76,66 segundos
Prueba 2	76,66 segundos
Prueba 3	86,39 segundos
Prueba 4	76,79 segundos

Il·lustració 38: Tabla de tiempos medios piezas en el proceso (20 segundos)

Se puede observar que los dos escenarios donde las piezas pasan un menor tiempo en el proceso son Prueba 1 (76,66 seg) y Prueba 2 (76,66seg). Cualquiera de estos dos escenarios son correctos para poder implementarlos. Se escoge indistintamente la Prueba 2 para ser programada.

5.4 Estudio de la flexibilidad del sistema escogido

Una vez obtenido el mejor de los escenarios propuestos (Prueba 2), se propone una serie de variaciones en la producción del proceso, con el objetivo de conocer cómo afectan al sistema.

Se realiza el estudio en el caso de que haya necesidad de aumentar la producción del proceso, aumentando el número de lotes de unidades o piezas. Este cambio se producirá si en la simulación el tiempo de llegada de piezas al proceso pasa a ser de cada 15 segundos.

Los tiempos en segundos de permanencia de las piezas en el proceso se muestran en la tabla siguiente:

Nombre simulación	Tiempo medio de permanencia de las piezas en el proceso si la llegada de piezas es cada 15 segundos
FIFO	171,79 segundos
Prioridad NANR	171,80 segundos
Prioridad SANR	171,80 segundos
Prueba 1	171,75 segundos
Prueba 2	171,75 segundos
Prueba 3	460,31 segundos
Prueba 4	171,76 segundos

Ilustración 39: Tabla de tiempos medios piezas en el proceso (15 segundos)

Se puede ver como las dos mejores opciones continúan siendo las mismas. Se puede afirmar que nuestro sistema se adapta a las fluctuaciones en la demanda, aspecto indispensable teniendo en cuenta el contexto de la industria 4.0 comentado en el apartado 1.3 Background.

6. Implementación de la solución óptima

6.1 Descripción del sistema experimental

Las instalaciones del laboratorio de la universidad donde se va a realizar la implementación de la solución óptima son:



Ilustración 40: Vista general célula flexible

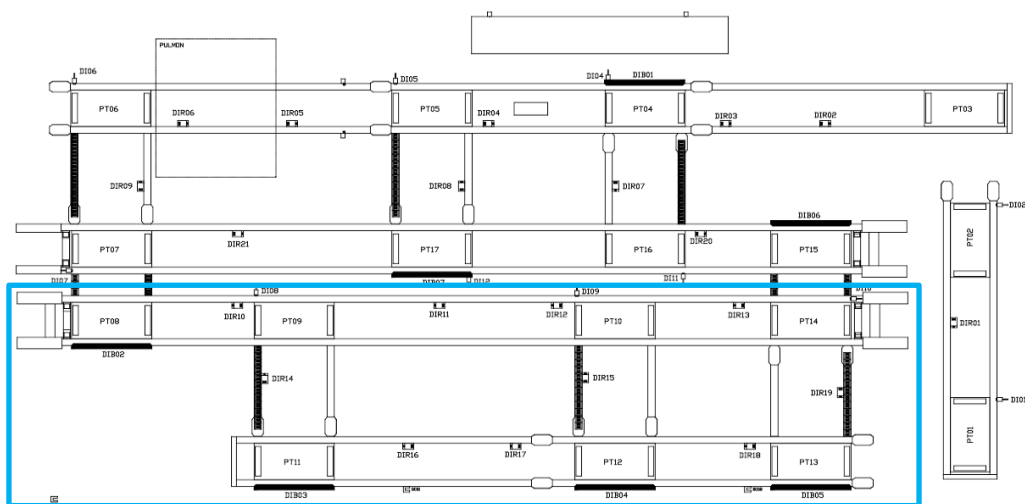


Ilustración 41: Instalaciones laboratorio 1

Para el caso de estudio se hace uso únicamente de la Línea Profibus del laboratorio, encuadrada en azul en la ilustración 41.

Cada pieza se desplaza por el sistema sobre una bandeja de un material férreo.

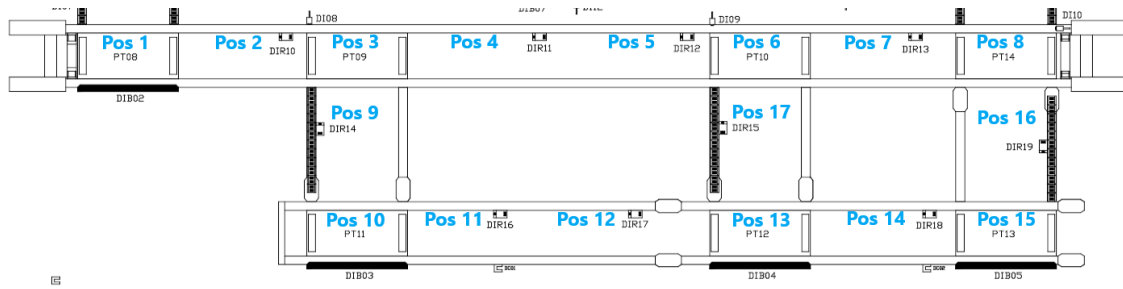


Ilustración 42: Instalaciones laboratorio 2

En la Línea Profibus hay catorce sensores, diez de los cuales cuentan también con retenedores.

Estos retenedores tienen como función permitir o bloquear el paso de las bandejas. Se activan con electroválvulas neumáticas.

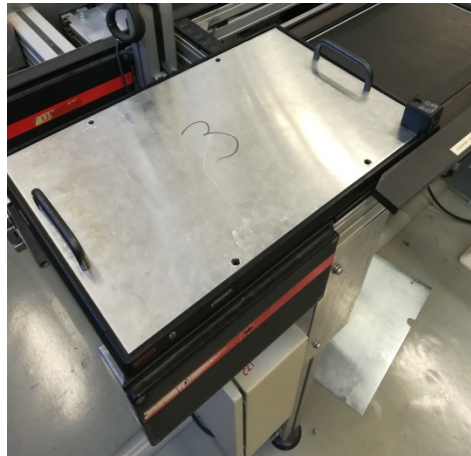
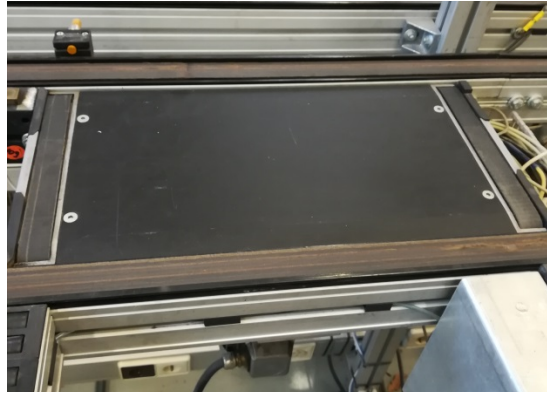


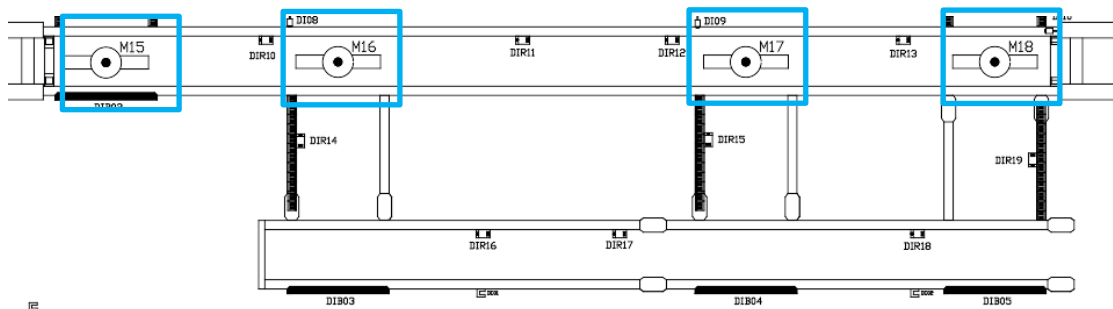
Ilustración 43: Bandeja

También hay plataformas. Su función es, como la de los retenedores, retener las piezas. Estas plataformas tienen dos cilindros neumáticos: uno que se encarga de la subida de la plataforma y otro que se encarga de la bajada. Cada uno de estos cilindros está gestionado por una electroválvula neumática.



Il·lustració 44: Plataforma del laboratori

La línia Profibus cuenta con cuatro motores asíncronos para la cinta transportadora. Se encuentran conectados directamente a red mediante un contactor de control.



Il·lustració 45: Motores cinta transportadora laboratorio

6.2 Diagrama de flujo del proceso

El proceso tiene diecisiete posiciones diferentes con dos bifurcaciones. Dentro de estas diecisiete posiciones se encuentran las dos etapas de mecanizado: el agujereado en la posición 11 y el taladrado en la posición 14.

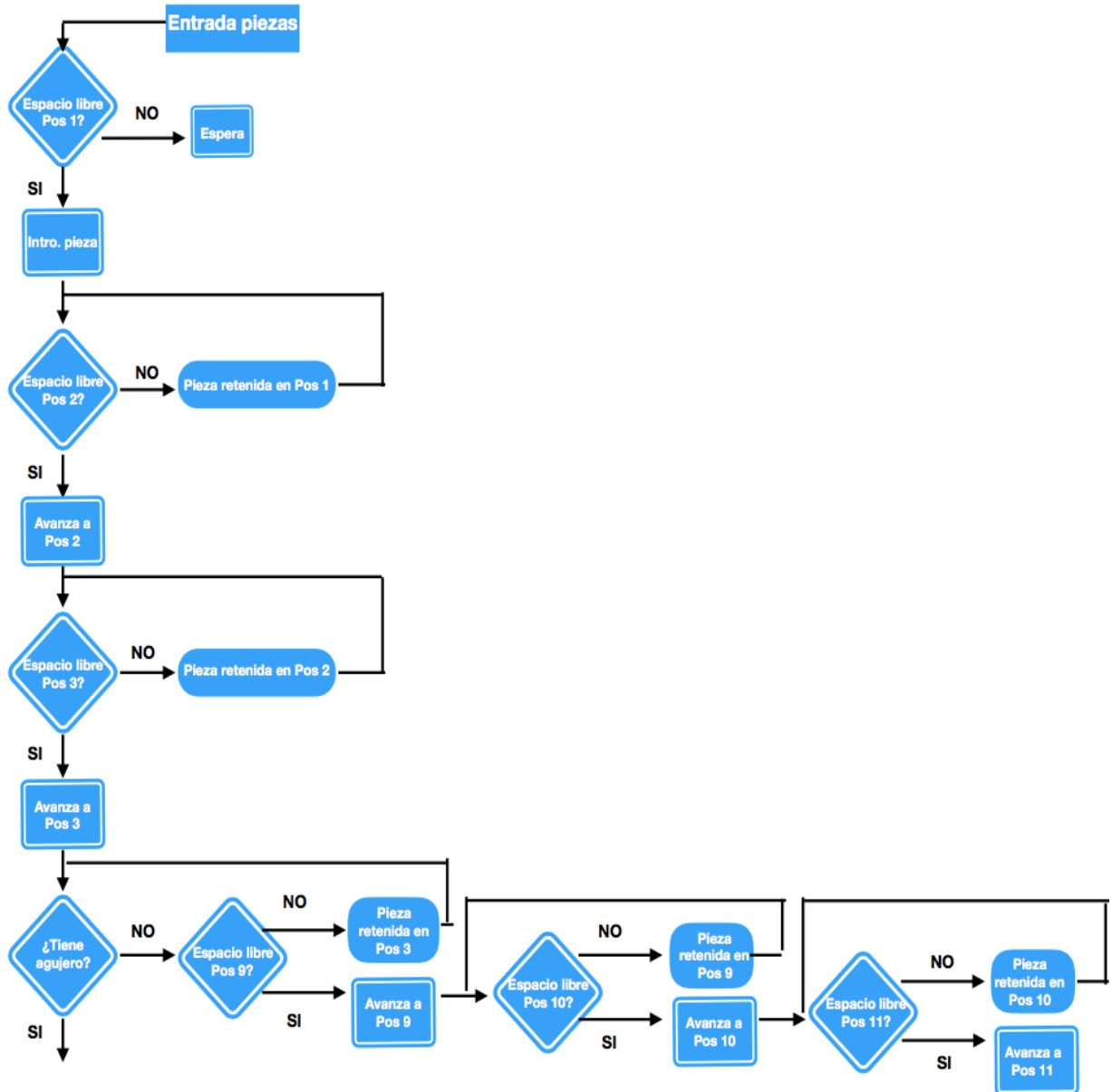
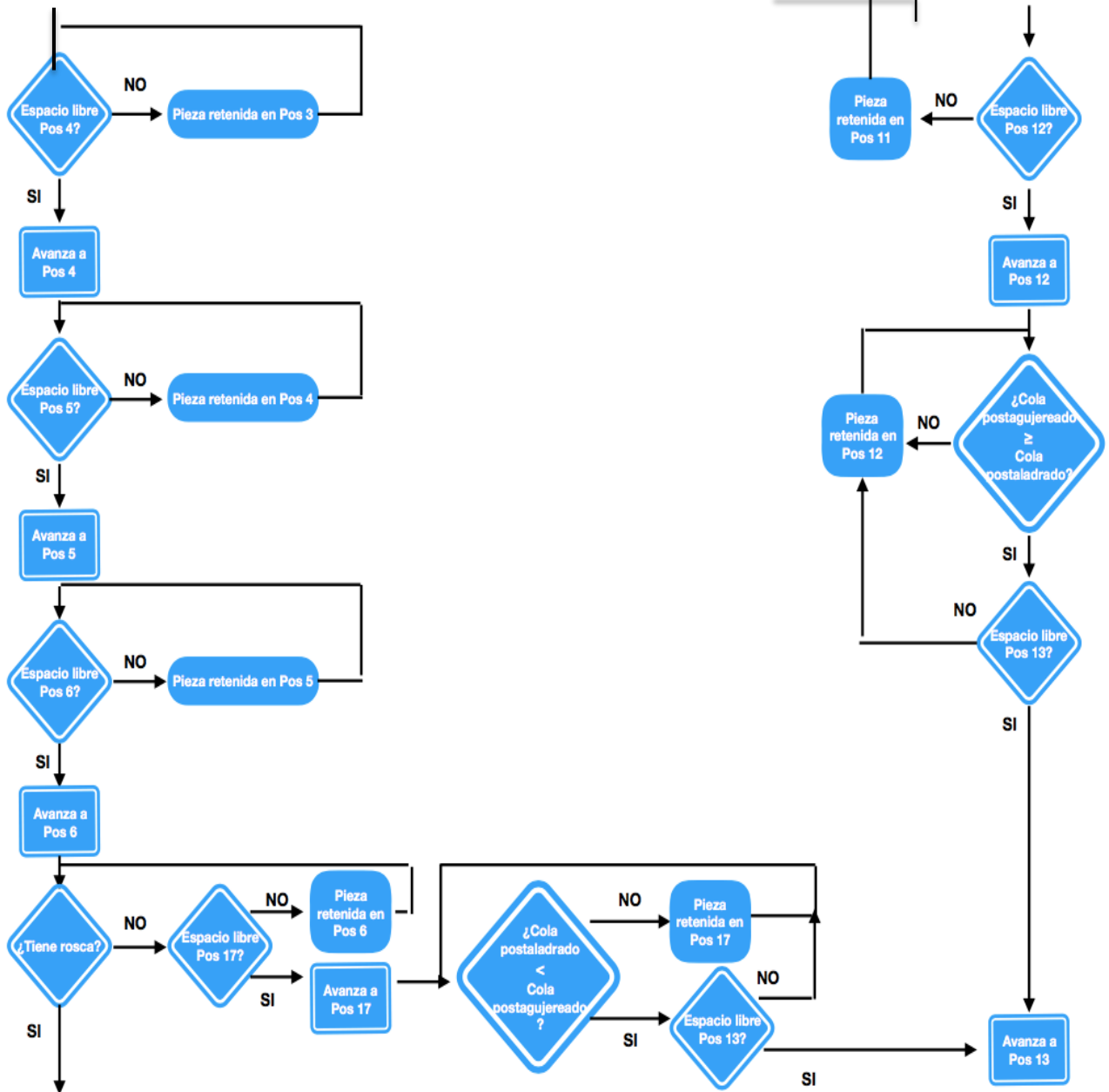
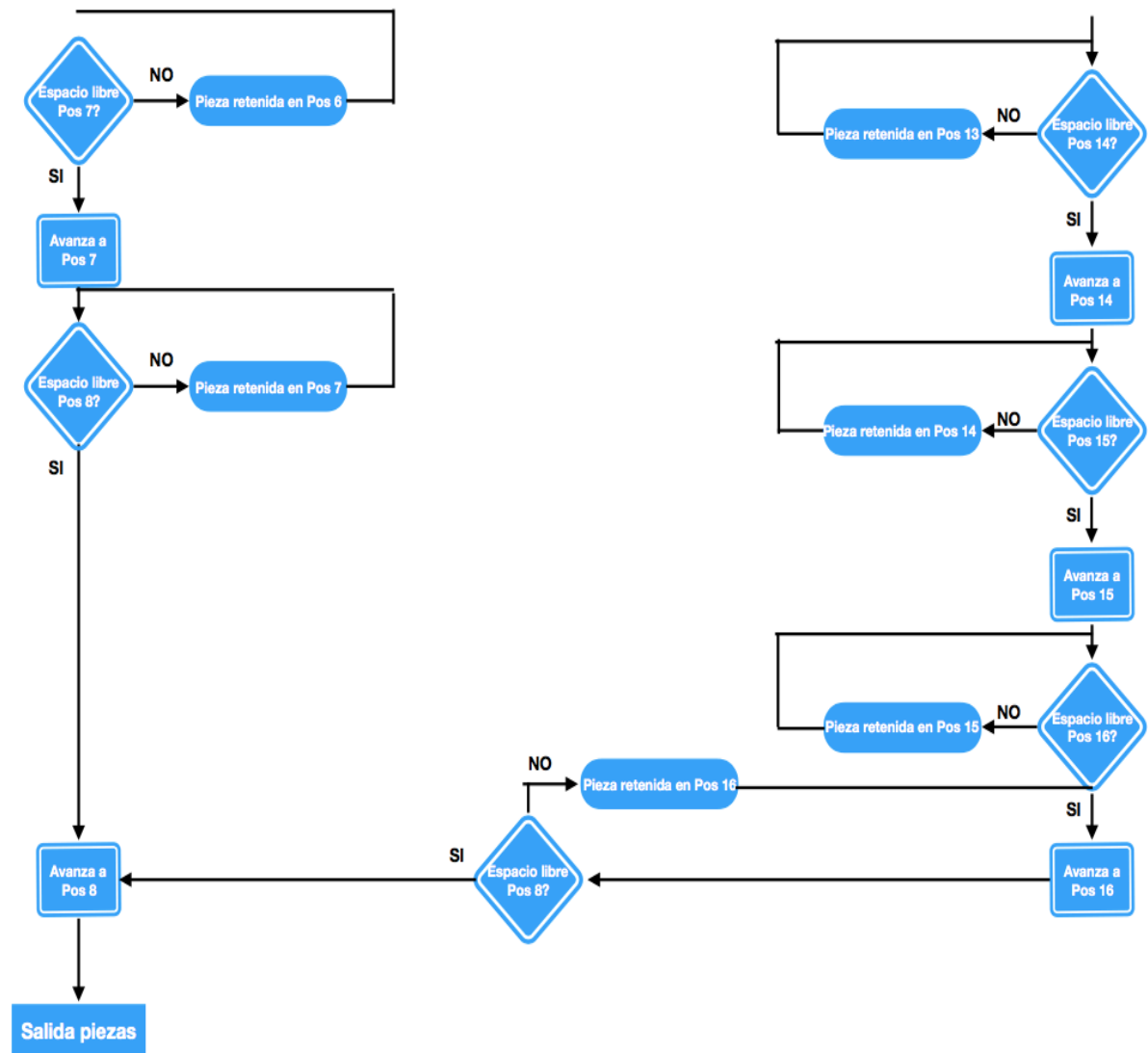


Ilustración 46: Diagrama de flujo del proceso 1



Il·lustració 47: Diagrama de flujo del proceso 2



Il·lustració 48: Diagrama de flujo del proceso 3

6.3 Configuración del programa

Para implementar esta solución en el PLC M340 de Schneider Electric se utiliza el programa Unity Pro XL.

Para configurar este programa se deben seguir los siguientes pasos:

- **Configuración inicial**

Al iniciar el programa se debe especificar el rack o bastidor del que se dispone, juntamente con el modelo de cada bloque. De forma automática aparece para escoger la gama del autómatas y la CPU o procesador del sistema. Para configurar el resto de módulos, se debe entrar en configuración, 0:Bus PLC. Haciendo doble click sobre cada módulo se abre un catálogo donde aparecen las diferentes opciones.

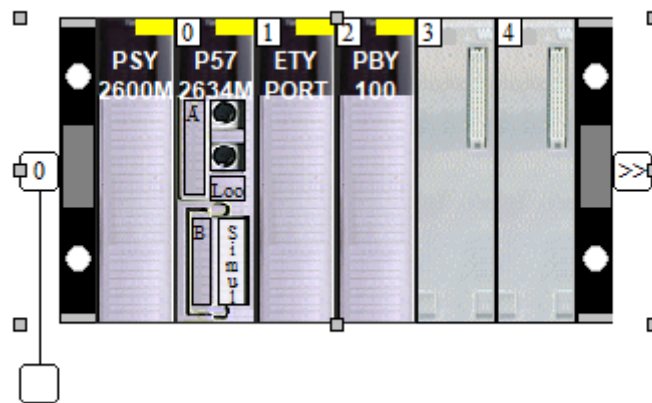


Ilustración 49: Configuración del Rack o bastidor en Unity Pro XL

- **Edición del código**

Para introducir el código Unity Pro XL permite crear varias secciones, donde en cada sección se programan cosas distintas y que luego al realizar el Ciclo de SCAN se ejecutan por orden.

Dispone de una barra de acceso rápido donde aparecen los diferentes componentes necesarios cuando se programa en lenguaje Ladder. Es tan simple como seleccionar el elemento deseado y arrastrarlo hasta la posición de la sección deseada. Una vez colocado, se debe asignar un nombre, el tipo y la dirección de la variable.

Un ejemplo de posible dirección es %Q0.3.2. Q indica que es una salida, 0 es el número de bastidor, 3 es la posición que ocupa este módulo y 2 es el canal correspondiente. En lugar de Q también se puede encontrar %I (entradas). Otra posibilidad es utilizar variables internas, en este caso se configura como %MX, siendo X el número que le se le atribuye a la variable interna.

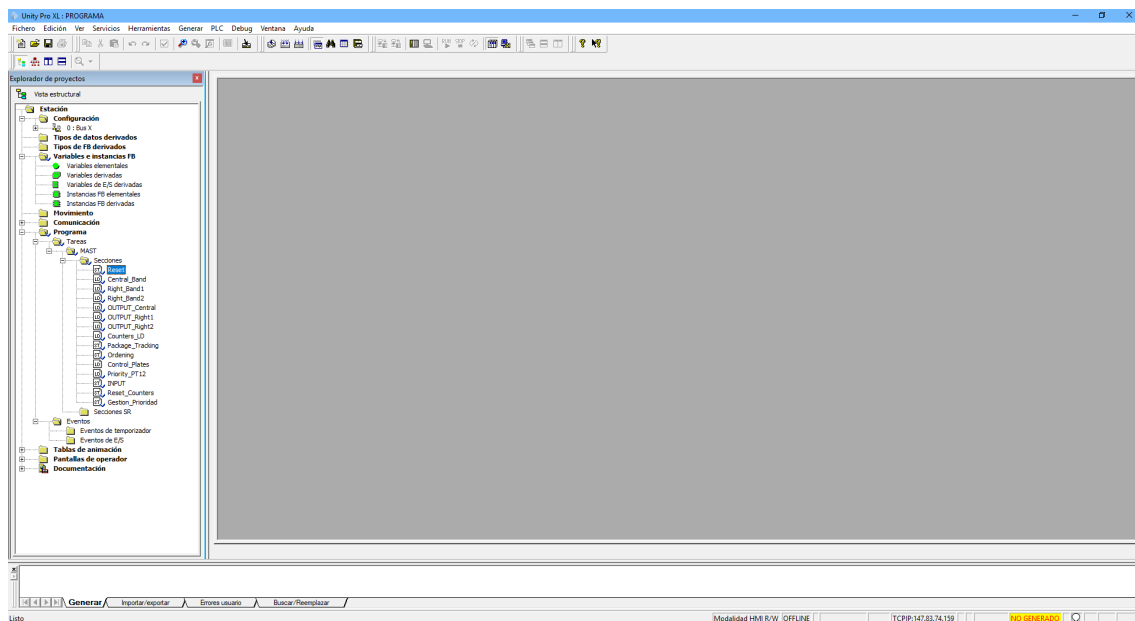


Ilustración 50: Pantalla principal Unity Pro XL

- **Cargar el programa**

Una vez finalizada la programación, se ejecuta el programa. Si la programación es correcta y se ha cargado el programa en el PLC, ya se puede poner en funcionamiento el sistema.

6.4 Estructura del programa

Para la programación se ha utilizado tanto lenguaje tipo Ladder como texto estructurado.

El programa se divide en diferentes secciones:

- Reset → en esta sección se resetean las variables necesarias para que al iniciar un nuevo ciclo de SCAN estén en el estado lógico inicial.
- Central_Band → gestiona el movimiento de las piezas cuando se encuentran en la zona central. En el caso de las bifurcaciones, gestionan el movimiento cuando las piezas se dirigen a una posición que está dentro del recuadro.

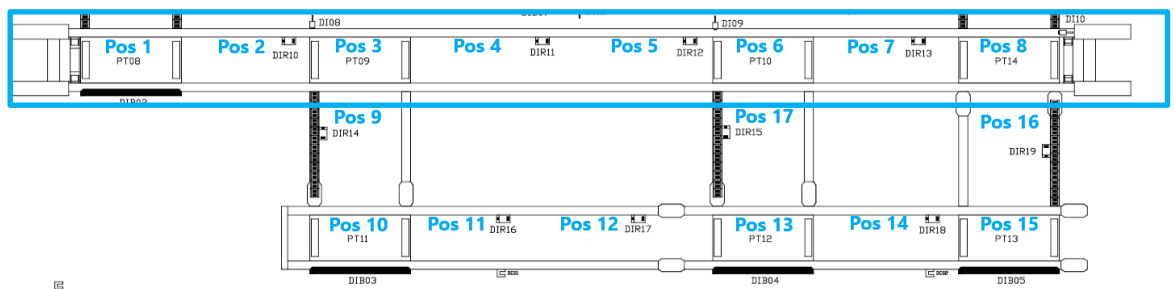


Ilustración 51: Instalaciones laboratorio 2

- Right_Band1 → gestión del movimiento de las piezas cuando se encuentran en las posiciones 3, 9, 10, 11 y 12. En el caso concreto de las piezas que se encuentran en la posición 3, solamente gestionan el movimiento cuando estas se dirigen a la posición 9.
- Right_Band2 → gestión del movimiento de las piezas cuando se encuentran en las posiciones 6, 17, 13, 14, 15 y 16. En el caso concreto de las piezas que se encuentran en la posición 6, solamente gestionan el movimiento cuando estas se dirigen a la posición 17.
- OUTPUT_Central → gestión de las plataformas que se encuentran dentro de la sección Central_Band.
- OUTPUT_Right1 → gestión de las plataformas que se encuentran dentro de la sección Right_Band1.
- OUTPUT_Right2 → gestión de las plataformas que se encuentran dentro de la sección Right_Band2. Además, también se gestiona la plataforma de la posición 15.
- Counters_LD → control y gestión de los diferentes contadores.
- Package_Tracking → control del tipo de pieza en las intersecciones.
- Ordening → gestión movimiento en bifurcaciones.
- Control_Plates → con esta sección es posible decidir el tipo de recorrido que hace la pieza: el propio de una pieza NANR, el de una pieza SANR o el de una pieza finalizada.

- Priority_PT12 → también se gestiona el movimiento en las bifurcaciones.
- INPUT → con esta sección se definen todas las piezas que entran al proceso.
- Reset_counters → gestión del reinicio de los contadores.
- Gestion_Prioridad → se gestiona la prioridad del punto crítico, correspondiente a la posición 13.

En el anexo se encuentra el código completo. En este apartado se van a exponer algunas partes relevantes del programa, facilitando así la comprensión global de este.

- **Gestión prioridad**

En esta ilustración se aprecia el código, en texto estructurado, utilizado en la sección de programa Gestion_Prioridad en que se gestiona la prioridad de paso en el punto crítico previamente explicado en el apartado 5.2.

```
IF (PTI10_MOVE_right AND DIRI15_READY) THEN
    IF ( C3_DIR11.CV- (C4_PT10.CV-C5_DIR13.CV) > (C2_PT09.CV-C3_DIR11.CV)-C7_DIR17.CV) THEN
        Priority_SANR := TRUE ;
        Priority_NANR := FALSE ;
    ELSE
        Priority_SANR := FALSE ;
        Priority_NANR := TRUE ;
    END_IF;
ELSE
    IF ( (C4_PT10.CV-C5_DIR13.CV) > C6_DIR14.CV-C7_DIR17.CV) THEN
        Priority_SANR := TRUE ;
        Priority_NANR := FALSE ;
    ELSE
        Priority_SANR := FALSE ;
        Priority_NANR := TRUE ;
    END_IF;
END_IF;
```

Ilustración 52: Captura sección Gestión_Prioridad Unity Pro XL

Se puede apreciar como el IF únicamente se ejecuta cuando se cumplen dos condiciones. Aclarar que este IF general se ha creado por las limitaciones de espacio del laboratorio, si se pudieran acumular piezas en la posición 17 no haría falta.

Es importante saber que para cada posición dentro de nuestro proceso se han creado tres variables diferentes:

1. Rest: hace referencia a que esa posición está en reposo.
2. Ready: hace referencia a que en esa posición hay una pieza.
3. Move: hace referencia a que la pieza que había en esa posición se está desplazando hacia la siguiente posición.

La primera de ellas es que la variable interna PTI10_MOVE_right esté activa. Esta variable hace referencia a la posición 6.

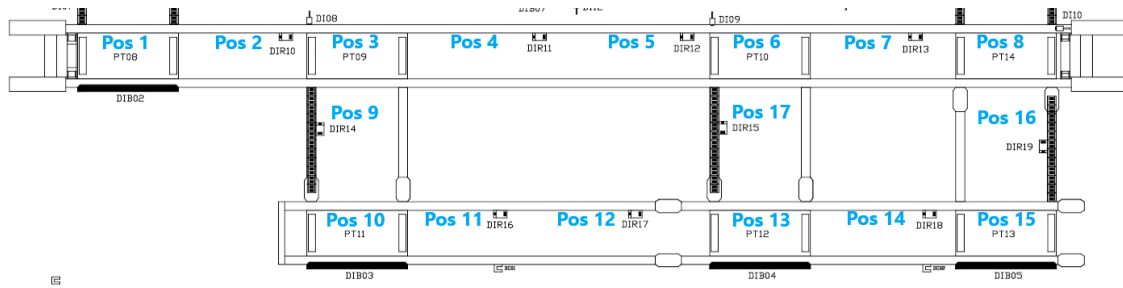


Ilustración 53: Instalaciones laboratorio 2

Cuando una pieza que esté en la posición 6, que se vaya a dirigir a la posición 17 (tenga agujereado pero no taladrado) y esté en condiciones de avanzar está variable PTI10_MOVE_right estará activa.

La variable DIRI15_READY hace referencia a que en la posición 17 el detector inductivo DIRI15 no detecte ninguna pieza y, por tanto, sea posible avanzar de la posición 6 a la 17.

Es entonces cuando se comparan el tamaño de las colas.

Para saber el tamaño de las colas es importante tener en cuenta que en cada una de las posiciones hay contadores. Por tanto, si hacemos la diferencia entre las piezas que han pasado por un punto en concreto y un punto posterior podemos encontrar las piezas que hay en cola.

Una vez dentro del primer IF, el siguiente IF que encontramos hace referencia al caso en que la cinta esté parada debido a que se han acumulado piezas en las posiciones 4, 5 y 6 como consecuencia de que una pieza con agujereado pero sin taladrado no puede avanzar. Si se cumple el requisito de tamaño de colas damos prioridad a las piezas SANR (Si Agujero No Rosca). Si no se cumple le atribuimos la prioridad a las piezas NANR (No Agujero No Rosca).

Si no se cumple el IF general se comparan el tamaño de colas para volver a determinar cuál debe tener prioridad.

Las variables Priority_SANR y Priority_NANR son las que dentro del programa se utilizan para activar las variables que hacen dirigirse a nuestras piezas por la línea central o desviarse por las bifurcaciones. Se puede apreciar en el siguiente apartado.

- **Prioridad en PT12**

En esta ilustración observamos una parte del código, en lenguaje Ladder, correspondiente a la sección Priority_PT12.

Las variables explicadas en el anterior apartado se utilizan para activar o desactivar la bobina. Si está activa la bobina asociada a la variable Priority_Mid2 son las piezas SANR las que tienen la prioridad. Mientras que si es la variable Priority_Right2 la que está activa, son las piezas NANR las que tienen la prioridad.

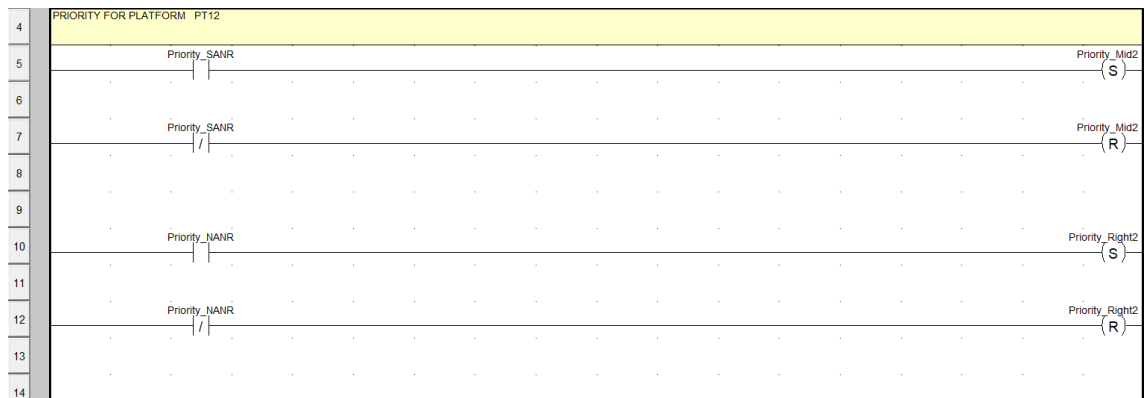
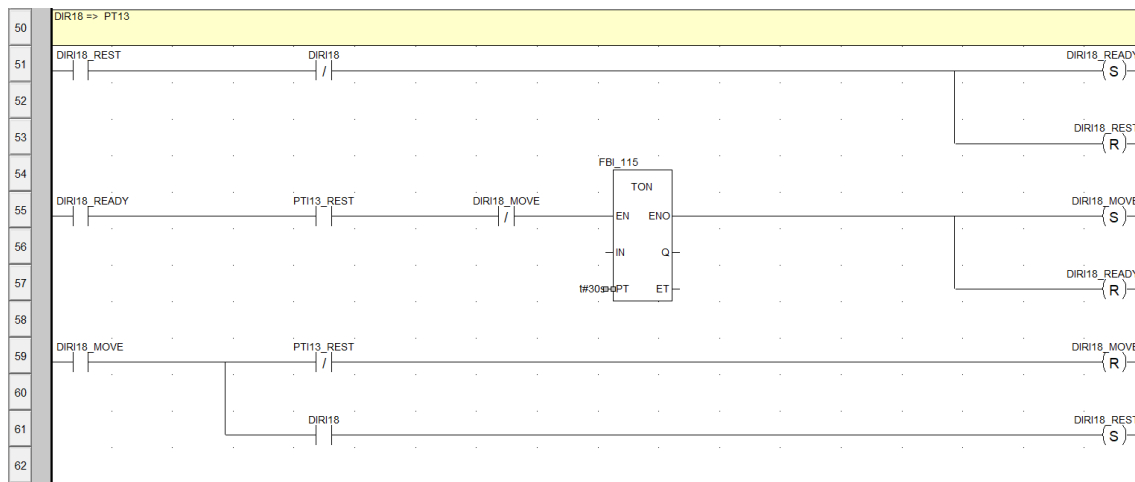


Ilustración 54: Captura sección Priority_PT12 Unity Pro XL

- **Gestión de desplazamientos**

En primer lugar se debe tener en cuenta que prácticamente se gestionan de la misma manera todos los desplazamientos dentro de nuestro proceso.

En esta ilustración observamos una parte del código correspondiente a la sección Right_Band2.



Il·lustració 55: Captura secció Right_Band2 Unity Pro XL

Podemos observar las condiciones que se deben dar para que una pieza avance desde la posición 14 a la posición 15.

En primer lugar se debe aclarar que el sensor DIRI18 es normalmente cerrado, esto significa que cuando detecte la bandeja su estado lógico será 0. Si está activa la variable rest asociada a la posición 14 (DIRI18_REST) entonces activamos la variable ready (DIRI18_READY) de esta posición y desactivamos la variable rest (DIRI18_REST).

Una vez activa la variable ready, si la variable rest de la posición 15 está activa (PTI13_REST), que esto implica que en la posición 15 ni hay pieza ni tampoco se está desplazando de la posición 15 a la posición 16 una pieza, y no está activa la variable move de la posición 14 (DIRI18_MOVE) se activa el temporizador.

Este temporizador es del tipo TON, que retrasa la conexión. Emula los 30 segundos que dura la operación de taladrado, para hacer la rosca.

Una vez pasados estos 30 segundos se activa la variable DIRI18_MOVE y se desactiva la DIRI18_READY.

Con esta variable activa, si la posición 15 no tiene activa su variable rest (PTI13_REST) se desactiva la variable DIRI18_MOVE. Y cuando el sensor DIRI18 deja de detectar la pieza vuelve a activar la variable DIRI18_REST.

6.5 Validación

Para poder comprobar la validez del sistema experimental se realizan una serie de pruebas. Estas pruebas las dividimos en funcionales y operacionales o de rendimiento.

- **Funcionales**

Son pruebas específicas y muy concretas para probar y validar que el software hace lo que se ha especificado. Se han realizado tres pruebas de este tipo:

Prueba 1	
Descripción del ensayo	Se va a introducir una pieza finalizada (con agujero y rosca) para comprobar que realiza el recorrido que debe
Resultado esperado del ensayo	La pieza recorre las posiciones: 1, 2, 3, 4, 5, 6, 7 y 8
Resultado obtenido del ensayo	La pieza recorre las posiciones: 1, 2, 3, 4, 5, 6, 7 y 8
Conclusiones del ensayo	El sistema funciona correctamente

Prueba 2	
Descripción del ensayo	Se va a introducir una pieza NANR (sin agujero ni rosca) para comprobar que realiza el recorrido que debe
Resultado esperado del ensayo	La pieza recorre las posiciones: 1, 2, 3, 9, 10, 11, 12, 13, 14, 15, 16 y 8
Resultado obtenido del ensayo	La pieza recorre las posiciones: 1, 2, 3, 9, 10, 11, 12, 13, 14, 15, 16 y 8
Conclusiones del ensayo	El sistema funciona correctamente

Prueba 3	
Descripción del ensayo	Se va a introducir una pieza SANR (con agujero sin rosca) para comprobar que realiza el recorrido que debe
Resultado esperado del ensayo	La pieza recorre las posiciones: 1, 2, 3, 4, 5, 6, 17, 13, 14, 15, 16 y 8
Resultado obtenido del ensayo	La pieza recorre las posiciones: 1, 2, 3, 4, 5, 6, 17, 13, 14, 15, 16 y 8
Conclusiones del ensayo	El sistema funciona correctamente

- **Operacionales o de rendimiento**

Son pruebas para probar y validar que el programa se comporta tal y como se ha simulado desde el punto de vista de rendimiento. Se han realizado dos pruebas diferentes:

Prueba 4	
Descripción del ensayo	Se van a introducir 5 piezas (3 finalizadas, 2 NANR y 1 SANR), una pieza cada 20 segundos.
Resultado esperado del ensayo	El tiempo medio que las piezas deben estar en el proceso, de acuerdo a la simulación con un lote de 5 piezas que se ha realizado en Arena, debe ser de 52,4 segundos.
Resultado obtenido del ensayo	El tiempo medio que permanecen las piezas en el proceso es de 52, segundos
Conclusiones del ensayo	El sistema funciona correctamente. La pequeña diferencia de tiempos es debida al error humano cronometrando los tiempos en el laboratorio.

Prueba 5	
Descripción del ensayo	Se van a introducir 10 piezas (5 finalizadas, 3 NANR y 2 SANR), una pieza cada 20 segundos.
Resultado esperado del ensayo	El tiempo medio que las piezas deben estar en el proceso, de acuerdo a la simulación con un lote de 10 piezas que se ha realizado en Arena, debe ser de 66,5 segundos.
Resultado obtenido del ensayo	El tiempo medio que permanecen las piezas en el proceso es de 66,8 segundos
Conclusiones del ensayo	El sistema funciona correctamente. La pequeña diferencia de tiempos es debida al error humano cronometrando los tiempos en el laboratorio.

7. Conclusiones

Utilizar el software Arena me ha hecho reflexionar y darme cuenta de la enorme importancia que tiene realizar un estudio de un proceso industrial. Es totalmente necesario para comprender el comportamiento que tendría en la realidad, creando un modelo de simulación y sin tener que realizar importantes inversiones. Desde mi punto de vista, creo que se debe poner mucho énfasis en este ámbito ya sea para diseñar un nuevo proceso o para plantear alternativas a alguno ya existente sin tener que paralizar la producción.

Además, se ha podido comprobar que el sistema se adapta a variaciones en la demanda. Esto es un aspecto fundamental y a tener mucho en cuenta debido al nuevo horizonte que presenta la industria 4.0. En el caso que, realizando la simulación, viéramos que nuestro proceso no es el óptimo únicamente debemos realizar unas pocas modificaciones en nuestro programa. El programa se ha diseñado teniendo en cuenta poder realizar las modificaciones de la forma más sencilla posible en caso de que sean necesarios los cambios.

El sistema es aplicable a cualquier línea de producción, incluso a la producción total. De esta manera, con mayor tiempo de estudio y programación este sistema se podría implementar para la total producción de la planta.

Este programa se ha diseñado para ser implementado en el laboratorio de la universidad. Las limitaciones físicas del laboratorio, especialmente respecto al espacio disponible, hacen que para aplicar el sistema en el ámbito real deban adaptarse parámetros como las colas o las distancias de las cintas. Con más espacio o con posibilidad de apilar las piezas hubiera sido más sencilla y eficiente la programación, ya que se reducirían el número de detenciones de la cinta.

En lo referente al ámbito personal ha sido una experiencia totalmente enriquecedora. Partía de un nivel nulo en simulación de procesos industriales y un nivel muy básico de automatización. A raíz del desarrollo de este trabajo me ha cambiado la forma de ver los procesos industriales, teniendo siempre muy presente el optimizar al máximo los recursos disponibles.

Para concluir, una posible continuación del trabajo sería el diseño de los elementos reales que conforman el proceso.

8. Presupuesto

Actividad	Duración (h)	Coste (€/h)	Coste total (€)
Diseño digital del proceso (entorno de simulación)	80	25	2000
Análisis de las simulaciones	20		500
Diseño del programa	25		625
Programación	130		3250
Validación y test	55		1375
TOTAL	310	25	7750

Ilustración 56: Presupuesto

9. Bibliografía

[1] Ehu.eus. *Sistemas de fabricación flexibles* [En línea]. Recuperada de:
<http://www.ehu.eus/manufacturing/docencia/1151_ca.pdf>

[2] *Simulación de Procesos: Softwares utilizados para la simulación de sistemas* [En línea]. Recuperada de:
<<https://simuluc.wordpress.com/2016/09/12/softwares-utilizados-para-la-simulacion-de-sistemas/>>

[3] Clarcat.com. *El software Arena*. [En línea]. Recuperada de :
<<https://www.clarcat.com/arena/>>

[4] Fabricantes-maquinaria-industrial.es. *¿Cómo alcanzar la automatización de procesos industriales?* [En línea]. Recuperada de:
<<https://www.fabricantes-maquinaria-industrial.es/alcanzar-la-automatizacion-procesos-industriales/>>

[5] Estampacionesjom.com. *Automatización industrial, estas son sus aplicaciones*. [En línea]. Recuperada de:
<<https://www.estampacionesjom.com/automatizacion-industrial-estas-son-sus-aplicaciones/>>

[6] Idboxrt.com. *Automatización industrial, la clave del éxito*. [En línea]. Recuperada de: <<https://idboxrt.com/blog/la-automatizacion-industrial-la-clave-del-exito-industrial/>>

[7] Ingmecafenix.com. *¿Qué es y para que sirve un PLC?* [En línea]. Recuperada de: <<https://www.ingmecafenix.com/automatizacion/que-es-un-plc/>>

[8] Automantenimiento.net. *Partes de un PLC*. [En línea]. Recuperada de:
<<http://automantenimiento.net/electricidad/partes-de-un-plc/>>

[9] Wikipedia.com/es. *Bus de campo*. [En línea]. Recuperada de:
<https://es.wikipedia.org/wiki/Bus_de_campo>

- [10] Infoplcn.net. El ciclo de SCAN en un autómata. [En línea]. Recuperada de: <<http://www.infoplcn.net/blogs-automatizacio/item/101930-ciclo-scan-automata-plc>>
- [11] Galia.fc.uaslp.mx. *Autómatas programables*. [En línea]. Recuperada de: <<http://galia.fc.uaslp.mx/~cantocar/automatas/PRESENTACIONES PLC PDF S/5 EL SCAN.PDF>>
- [12] Slideshare.net/es. *Sensores inductivos y PLC*. [En línea]. Recuperada de: <<https://es.slideshare.net/josueacerov/sensores-inductivos-y-plc>>
- [13] Ieec.uned.es. *Master Degree: Ingeniería de Sistemas Industriales*. [En línea]. Recuperada de: <<http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion de referencia ISE6 1 1.pdf>>
- [14] Unicrom.com. *Historia del PLC*. [En línea]. Recuperada de: <<https://unicrom.com/historia-del-plc-modicon-modbus/>>
- [15] Ingmecafenix.com. *Lenguajes para programación de PLC*. [En línea]. Recuperada de: <<https://www.ingmecafenix.com/automatizacion/lenguajes-programacion-plc/>>
- [16] Contaval.es. *Programación tipo Ladder*. [En línea]. Recuperada de: <<https://www.contaval.es/programacion-tipo-ladder/>>
- [17] Piedrafito Moreno, Ramón. *Ingeniería de la automatización industrial*. 2ª Edición ampliada y actualizada. Madrid: RA-MA, 2004. ISBN: 84-7897-604-3
- [18] Rubio Sánchez, Juan Luis. *Problemas de automatización industrial*. 1ª Edición (21 de enero de 2019). ISBN: 978-84-454-3794-0.
- [19] García i López, Óscar. *Treball Final de Grau: Estudi, disseny i aplicació d'un automatisme programat sobre una línia de producció*. UPC Terrassa: 10 de enero de 2019.

[20] Rodríguez Vilches, Rubén. *Trabajo Fin de Máster: Simulación de un proceso de fabricación*. Universidad Pública de Navarra: 15 de septiembre de 2017.